



## Memòria del projecte



Facultat d'Informàtica de Barcelona  
Universitat Politècnica de Catalunya

Treball realitzat per:  
**Raul Fernández García**

Director:  
**Jordi Escudero Subirana**

Ponent:  
**Llorenç Cerdà-Alabern**

Barcelona, 26 de Juny 2019

## Resum

Les tecnologies digitals han transformat la producció i la distribució del contingut multimèdia. Aquest mercat està previst que continuï creixent en els 5 anys següents, i actualment està valorat per sobre dels \$2 trilions.

BLOOMEN es planteja com una investigació i exploració dins del camp de les tecnologies *Blockchain*, per tal de desenvolupar un disseny basat en aquestes tecnologies i aplicat a diferents casos d'ús al sector del contingut multimèdia. En aquest projecte es desenvoluparà 3 pilots funcionals per diferent contingut multimèdia:

- Pilot WebTV
- Pilot Music
- Pilot Photos

La meua comesa serà desenvolupar diferents components pel Pilot WebTV i pel Pilot Music. Els components principals seran els *Smart Contracts* i una Interfície Web. El primer, contindrà tota la lògica del negoci, per exemple gestionarà els drets d'autor i tot el contingut multimèdia. El segon consistirà en una plataforma per crear, cercar i reclamar contingut multimèdia.

## Abstract

Digital technologies have transformed the production and distribution of media content. This market is expected to continue growing in the next 5 years, and is currently valued above \$ 2 trillion.

BLOOMEN is planned as a research and exploration within the field of Blockchain technologies, to develop a design based on these technologies and applied to different cases of use of the media content sector. In this project 3 functional pilots will be developed for different multimedia content:

- WebTV Pilot
- Music Pilot
- Photos Pilot

My goal will be to develop different components for the WebTV Pilot i for the Music Pilot. The main components will be the Smart Contracts and a Web interface. The first, will have all the business logic, for example management of the copyright and all media content. The second will consist of a platform to create, search and claim media content.

# ÍNDEX

<b>1. Context</b>	<b>8</b>
1.1 Actors implicats	8
<b>2. Estat de l'art</b>	<b>10</b>
<b>3. Formulació del Problema</b>	<b>12</b>
3.1 Objectius	12
<b>4. Abast</b>	<b>14</b>
4.1 Possibles obstacles	14
<b>5. Metodologia</b>	<b>15</b>
5.1 Mètodes de treball	15
5.2 Eines de seguiment	16
5.3 Mètode de validació	17
<b>6. Planificació general</b>	<b>18</b>
6.1 Planificació estimada del projecte	18
6.2 Alternatives i pla d'acció	18
6.3 Recursos	19
6.3.1 Recursos personals	19
6.3.2 Recursos materials	19
<b>7. Descripció de tasques</b>	<b>20</b>
<b>8. Calendari</b>	<b>21</b>
8.1 Estimació d'hores	21
<b>9. Gestió econòmica del projecte</b>	<b>22</b>
9.1 Introducció	22
9.2 Estimació de costos	22
9.2.1 Costos de recursos humans	22
9.2.2 Costos de recursos materials	24
9.2.3 Costos generals indirectes	24
9.2.4 Contingència	25
9.2.5 Imprevistos	25
9.2.6 Pressupost final	25
9.3 Control de gestió	26
<b>10. Sostenibilitat i compromís social</b>	<b>27</b>
10.1 Ambiental	27

10.2 Econòmic	27
10.3 Social	27
10.4 Matriu de sostenibilitat	28
<b>11. Disseny</b>	<b>29</b>
11.1 Pilot WebTV	29
11.1.1 App	30
11.1.2 Smart Contracts	31
11.1.3 Bloomen-wallet-cli	33
11.1.4 Service Demonstrator	33
11.2 Pilot Music	37
<b>12. Desenvolupament</b>	<b>40</b>
12.1 REST API	40
12.2 Smart Contracts	42
12.3 Interfície	53
12.4 Llibreria NPM	61
<b>13. Conclusions</b>	<b>66</b>
<b>14. Apèndixs</b>	<b>67</b>
14.1 Gantt	67
14.2 Repositoris	67
<b>15. Bibliografia</b>	<b>68</b>

## ÍNDEX DE FIGURES

Figura 1. *Arquitectura d'una aplicació descentralitzada ... 10*

Figura 2. *Avantatges i inconvenients del mètode en cascada ... 15*

Figura 3. *Fases del mètode en cascada ... 16*

Figura 4. *Workflow a Git ... 17*

Figura 5. *Descripció del Prototip ... 30*

Figura 6. *Arquitectura de l'aplicació ... 31*

Figura 7. *Diagrama dels Smart Contracts ... 31*

Figura 8. *Cas bàsic de Prepaid Cards ... 32*

Figura 9. *Menu Service Provider ... 33*

Figura 10. *Menu Service Costumer ... 33*

Figura 11. *Llista de demos ... 34*

Figura 12. *Llista de vídeos ... 34*

Figura 13. *Llista de vehicles ... 35*

Figura 14. *Llista d'articles per oficines ... 35*

Figura 15. *QRs per comprar i veure un vídeo ... 36*

Figura 16. *Reproducció d'un vídeo ... 36*

Figura 17. <i>Arquitectura del Pilot de Música</i> ...	37
Figura 18. <i>Pàgina inicial</i> ...	38
Figura 19. <i>main.ts de l'API</i> ...	40
Figura 20. <i>Portal Swagger</i> ...	41
Figura 21. <i>Codi del /wallet</i> ...	42
Figura 22. <i>Estructura projecte de Truffle</i> ...	43
Figura 23. <i>Comanda per compilar</i> ...	43
Figura 24. <i>Comanda per debuggar</i> ...	43
Figura 25. <i>Comanda per testejar</i> ...	44
Figura 26. <i>Interfície de Ganache</i> ...	45
Figura 27. <i>Interfície ERC20</i> ...	46
Figura 28. <i>Ntua Token Smart Contract</i> ...	46
Figura 29. <i>Ntua Token funcionalitats</i> ...	47
Figura 30. <i>Estructura media_content</i> ...	47
Figura 31. <i>Esdeveniments Uploading i Retrieving</i> ...	48
Figura 32. <i>Asset Manager funcionalitats</i> ...	48
Figura 33. <i>Estructura media_ownership i media_content</i> ...	49
Figura 34. <i>Mapping i createMedia</i> ...	50
Figura 35. <i>Copyright funcionalitat</i> ...	50

Figura 36. <i>Diagrama de seqüència del addCard ...</i>	51
Figura 37. <i>Diagrama de seqüència del buyCard ...</i>	52
Figura 38. <i>Diagrama de seqüència de validateCard ...</i>	53
Figura 39. <i>Pantalla inicial ...</i>	54
Figura 40. <i>Resultat d'una cerca ...</i>	54
Figura 41. <i>Claims tab ...</i>	55
Figura 42. <i>Tasks tab ...</i>	55
Figura 43. <i>Botó per afegir nous assets ...</i>	56
Figura 44. <i>Formulari per afegir un nou asset ...</i>	56
Figura 45. <i>Adreça seleccionada ...</i>	57
Figura 46. <i>Cerca d'una cançó ...</i>	57
Figura 47. <i>Crear un claim d'una cançó ...</i>	58
Figura 48. <i>Formulari descripció del claim ...</i>	59
Figura 49. <i>Descripció del claim ...</i>	59
Figura 50. <i>Tasks tab amb nova task ...</i>	60
Figura 51. <i>Claim tab amb nou claim ...</i>	61
Figura 52. <i>Constructor ...</i>	61
Figura 53. <i>Creació d'un JsonPathPair ...</i>	62
Figura 54. <i>Valor del JsonPathPair ...</i>	62

Figura 55. *Mètode marshall* ... 62

Figura 56. *Transformar Json en un JsonPath* ... 63

Figura 57. *Valor del JsonPath* ... 63

Figura 58. *Mètode unMarshall* ... 63

Figura 59. *Valor del JsonPath* ... 64

Figura 60. *Transformar JsonPath en un Json* ... 64

Figura 61. *Mètode compareJsonPath* ... 64

Figura 62. *Comparar 2 Json's* ... 65

Figura 63. *Valor de la comparació* ... 65

Figura 64. *Diagrama de Gantt* ... 67



## 1. Context

Aquest projecte és un Treball de Final de Grau de l'especialitat de Tecnologies de la Informació de la Facultat d'Informàtica de Barcelona dirigit per Jordi Escudero Subirana i amb ponència de Llorenç Cerdà-Alabern. Es tracta d'un projecte de modalitat B desenvolupat pel Centre de Competències de Mobilitat a l'empresa Worldline Iberia S.A.

BLOOMEN<sup>[1]</sup> es planteja com una investigació i exploració dins del camp de les tecnologies *Blockchain*, per tal de desenvolupar un disseny, basat en aquestes tecnologies, aplicat a diferents casos d'ús al sector del contingut multimèdia.

El que implementaré d'aquest projecte estarà dividit, principalment, en dues parts:

- 1) Una plataforma per crear, compartir i utilitzar contingut multimèdia. Aquesta plataforma oferirà als usuaris la capacitat de crear contingut, per exemple una cançó. A més, també oferirà als usuaris la possibilitat de cercar nou contingut i de demanar l'ús de tot el contingut del qual no en sigui propietari.
- 2) Blockchain<sup>[2]</sup>. Constarà d'un conjunt de *Smart Contracts*<sup>[3]</sup> que s'encarregaran d'aplicar unes normes i conseqüències, com faria un document legal tradicional.

### 1.1 Actors implicats

En el context d'aquest projecte hi ha una sèrie d'actors implicats. En aquest apartat es defineixen els principals actors que estan implicats de manera directa o indirecta al projecte.

- 1) Director, ponent i programador. Són les persones que participaran del projecte i del seu avanç fins que aconsegueixi els objectius establerts en el temps adjudicat pel Treball de Final de Grau. El director Jordi Escudero Subirana i el ponent Llorenç Cerdà-Alabern tenen la funció de guiar al programador durant tota la trajectòria del projecte. El programador, Raul Fernández García, serà qui s'encarregarà d'implementar i dur a terme tot el planificat al Treball de Final de Grau.



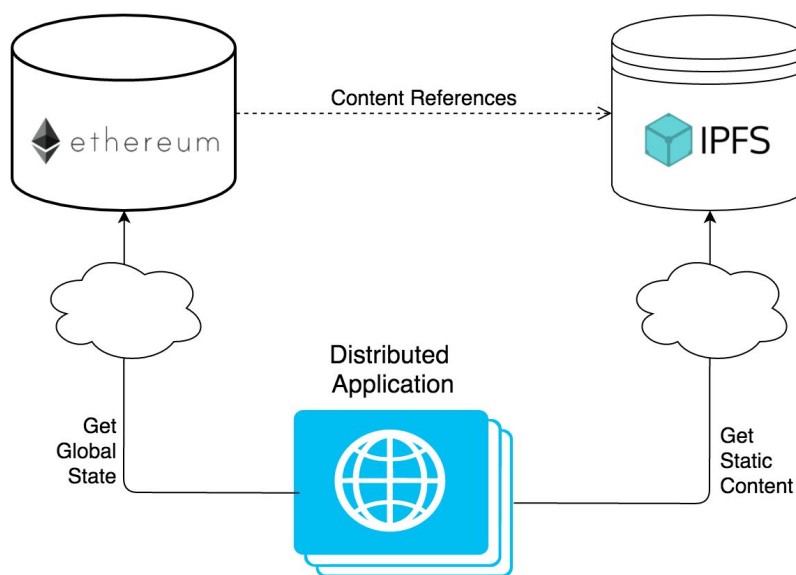
- 2) Creadors de contingut. Són les persones a les quals beneficiarà aquest desenvolupament, ja que podran donar d'alta contingut, i gestionar els seus drets d'autor.

## 2. Estat de l'art

Aquest projecte no és l'únic que incideix en la idea d'integrar la tecnologia Blockchain amb la gestió del contingut multimèdia. A continuació, esmentaré alguns projectes actuals que hi ha en funcionament:

- Hubii<sup>[4]</sup>. És un mercat de contingut basat en la Blockchain. Està obert per qualsevol creació de contingut, el seu objectiu principal és treure els intermediaris i donar el poder als creadors i usuaris.
- Steemit<sup>[5]</sup>. És una plataforma basada en Blockchain de *social media* a on qualsevol persona pot guanyar recompenses. Va ser una de les primeres plataformes a centrar-se en la descentralització del contingut i la seva creació.

La solució que la comunitat<sup>[6]</sup> proposa consisteix en una aplicació distribuïda amb la següent arquitectura:



*Figura 1. Arquitectura d'una aplicació descentralitzada*

Aquesta arquitectura la farem servir al projecte, on podem veure que l'aplicació es comunicarà amb la Blockchain, en aquest cas Ethereum<sup>[7]</sup>, que mostrarà els estats globals de les estructures de dades.

Aquesta solució resol un dels problemes principals de Ethereum: Reduir la quantitat de dades emmagatzemades a la Blockchain. Com tot el contingut audiovisual cada vegada tendeix a ocupar més i més, és necessari algun tipus de tecnologia o sistema que ens faciliti l'emmagatzematge i l'accés a les dades sense repercutir negativament al funcionament de la Blockchain. Aquesta tecnologia és la IPFS<sup>1</sup>.

En conclusió, agafarem aquest disseny com a disseny inicial pel projecte, ja que així, sense veure les possibles futures complicacions, sembla que s'adequa a les necessitats del projecte.

---

<sup>1</sup> InterPlanetary File System, és un protocol i xarxa per emmagatzemar contingut de manera distribuïda

### 3. Formulació del Problema

Les tecnologies digitals han transformat la producció i la distribució del contingut multimèdia. Aquest mercat està previst que continuï creixent en els 5 anys següents, i actualment està valorat per sobre dels \$2 trilions<sup>[8]</sup>.

Hi ha molts reptes que dins d'aquesta indústria encara es mantenen sense resoldre, especialment relacionats amb la manera en què el contingut digital pot ser copiat i distribuït lliurement a internet, i com els creadors de contingut són recompensats quan el seu material està sent utilitzat o comprat a través de canals legítims.

La tecnologia Blockchain i les divises virtuals poden donar resposta a la idea anteriorment esmentada, preservant la privacitat, tenint taxes petites, transaccions financeres instantànies i sense intermediaris.

El principal objectiu de Bloomen és estendre l'ús de la tecnologia Blockchain per gestionar diferents transaccions entre usuaris en línia, proveir una manera innovadora de crear, compartir i consumir contingut, monetitzacions i *copyrighting*.

En particular en aquest projecte, Blockchain serà utilitzada com a base de dades distribuïda per la informació del *copyright* del contingut, per fer ràpid micropagaments sobre media content, i per la transparència en la gestió i monetització del *copyright*.

#### 3.1 Objectius

L'objectiu principal d'aquest projecte és crear tant la plataforma com la Blockchain, prèviament esmentades a l'apartat de context.

La plataforma comptarà amb una sèrie de funcionalitats bàsiques cobertes:

1. Una secció *Home* que ens permet cercar i generar recursos, i una secció *Claim*<sup>2</sup> que ens permet gestionar les peticions d'ús dels nostres continguts.
2. Desplegar un menú per la creació de moneders (en anglès, *wallets*) i seleccionar-ne un.
3. Mostra l'adreça del moneder en ús.

---

<sup>2</sup> Reclamació



A més, un altre objectiu del projecte consisteix a integrar diferents tecnologies.

Un objectiu personal dins d'aquest projecte és aprendre diferents tecnologies noves i nous llenguatges per mi, com per exemple Solidity<sup>[9]</sup> en Ethereum en la construcció dels *Smart Contracts* de la Blockchain.

## 4. Abast

L'abast d'aquest projecte és aconseguir desenvolupar un *proof-of-concept*<sup>3</sup> que sigui usable amb les idees prèviament esmentades.

Aprofitar aquest estudi per tal de treure conclusions respecte a la viabilitat d'aprofitar la Blockchain i així construir per tota la indústria del contingut multimèdia un nou concepte descentralitzat.

La meta final és implementar amb èxit les dos grans parts del projecte:

- 1) La plataforma per crear, compartir i utilitzar contingut.
- 2) Blockchain.

A més, no només implementar aquestes parts per separat, sinó aconseguir que ambdues parts estiguin ben integrades i, per tant, que treballin conjuntament de la manera planificada.

### 4.1 Possibles obstacles

En aquest apartat expressaré els possibles obstacles que em puc trobar durant el desenvolupament del projecte.

Els obstacles que poden aparèixer durant el desenvolupament del projecte els classificaré en dues parts segons la seva causa:

- 1) La intenció d'integrar múltiples tecnologies i que treballin conjuntament representa un possible risc. Aquest risc pot afectar sobretot a la planificació inicial, produint variacions en el que s'esperava.
- 2) Per implementar la Blockchain utilitzaré el llenguatge de programació Solidity. Aquest llenguatge no porta gaire temps i per tant, en estar en una fase de beta i no haver-hi gaire documentació per part de la comunitat això podria derivar en algunes situacions costoses de resoldre.

---

<sup>3</sup> Proof-of-concept, és una prova de concepte

## 5. Metodologia

En aquest apartat es descriuran els mètodes de treball, les eines de seguiment i els mètodes de validació que s'utilitzaran durant el desenvolupament del treball.

### 5.1 Mètodes de treball

Pel desenvolupament d'aquest projecte s'utilitzarà la metodologia en cascada<sup>[10]</sup>. Aquesta és un model SDLC<sup>4</sup> en el que el procés de desenvolupament passa per les fases d'anàlisis, projectar, realitzar, fer tests, implementar i fer manteniment. Aquest model SDLC inclou una execució gradual de cada fase. Aquest procés és estrictament documentat i predefinit amb característiques esperades per cada fase del cicle de vida del desenvolupament de *software*.

ADVANTAGES	DISADVANTAGES
Simple to use and understand	The software is ready only after the last stage is over
Management simplicity thanks to its rigidity: every phase has a defined result and process review	High risks and uncertainty
Development stages go one by one	Not the best choice for complex and object-oriented projects
Perfect for the small or mid-sized projects where requirements are clear and not equivocal	Inappropriate for the long-term projects
Easy to determine the key points in the development cycle	The progress of the stage is hard to measure while it is still in the development
Easy to classify and prioritize tasks	Integration is done at the very end, which does not give the option of identifying the problem in advance

*Figura 2. Avantatges i inconvenients del mètode en cascada*

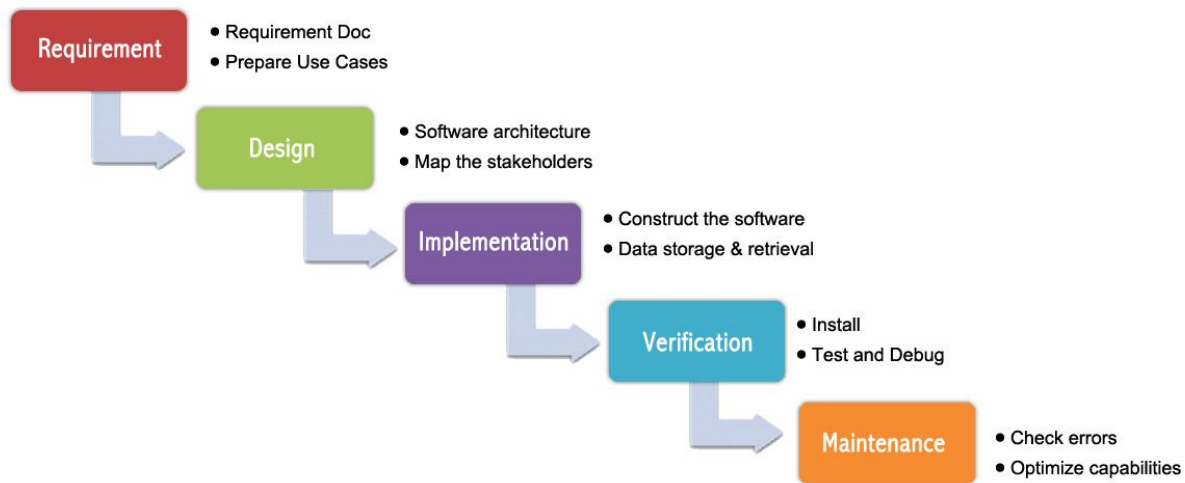
Casos d'ús del model en cascada:

- Els requeriments estan documentats de manera precisa.
- La definició del producte és estable.
- El conjunt de tecnologies emprades està predefinit el que fa que no sigui dinàmic.

<sup>4</sup> Software Development Life Cycle



- No hi ha requeriments ambigus.
- El projecte és curt.



*Figura 3. Fases del mètode en cascada*

## 5.2 Eines de seguiment

Per a la planificació de les tasques i la gestió del projecte s'utilitzarà JIRA<sup>[11]</sup> Software. Aquest ens permet fer supervisions de les incidències i del projecte de forma flexible.

Per la gestió del codi s'utilitzarà un software de control de versions per a poder portar un exhaustiu control del contingut. Aquest software escollit serà Git<sup>[12]</sup>.

A més de Git, també utilitzaré Github. Aquesta és una plataforma de desenvolupament col·laboratiu a on es poden allotjar projectes de Git.

La manera en què utilitzaré Git és seguint el següent Git *workflow*:

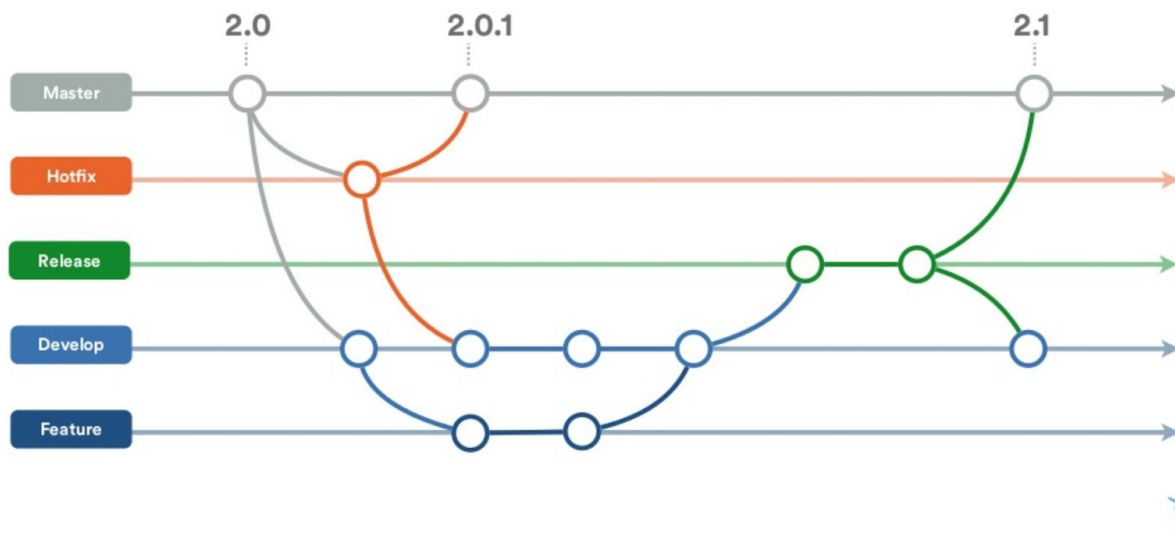


Figura 4. Workflow a Git

### 5.3 Mètode de validació

Per tal de validar la feina feta, hi haurà una comunicació fluida i habitual amb el director del TFG. Ell s'encarregarà de comprovar que tot avanci com estava planificat. Tot això es farà a partir de validacions parcials de l'evolució del projecte. Aquestes validacions prenen la forma de reunions periòdiques per tal de detectar qualsevol desviació, encara que sigui mínima, dels objectius del projecte.

## 6. Planificació general

### 6.1 Planificació estimada del projecte

Aquest projecte té una durada aproximada d'uns 4 mesos, amb inici el 18 de Febrer i final el 24 de Juny. L'estimació de la càrrega de treball que comportarà és d'unes 450 hores, aproximadament. Dins d'aquestes 450 hores estan incloses totes les tasques a realitzar del projecte.

### 6.2 Alternatives i pla d'acció

Durant el transcurs del desenvolupament del projecte, hi pot haver principalment una desviació, sent aquesta causada per una mala planificació del temps.

Pot haver-hi múltiples raons per la qual pot esdevenir una desviació temporal.

- Una desviació en el temps dedicat a aprendre el funcionament de les noves tecnologies.
- Sobrepassar el temps planificat.
- Finalitzar la tasca en menys temps.

Si en un d'aquests moments existeix una desviació de més de 20 hores en la realització d'una tasca, s'aplicaran mesures correctives i es replanificarà. A més, la llista de tasques a desenvolupar estan en ordre de prioritat, per tant en cas de manca de temps les tasques afectades seran les menys prioritàries.

Les mesures correctives que s'aplicaran són:

- 1) Cancel·lar el cas d'ús que s'estava desenvolupant.
- 2) Modificar el cas d'ús per tal que sigui més senzill.
- 3) Cancel·lar un cas d'ús menys prioritari per tal d'invertir més hores en un altre.

Amb aquestes mesures correctives s'assegura que el projecte es finalitzarà.

## 6.3 Recursos

Els recursos previstos per la realització d'aquest projecte són els següents:

### 6.3.1 Recursos personals

L'únic recurs personal de què es disposarà per la realització del treball serà d'un treballador en pràctiques. Aquest treballador realitzarà múltiples funcions per poder finalitzar totes les tasques necessàries per dur a terme el projecte.

El temps de dedicació del treballador al projecte serà d'unes 25 hores setmanals. Podrà variar una mica segons la càrrega de treball que tingui. A més, hi haurà setmanes en les quals, per falta de més temps, s'hauran de realitzar més d'aquestes 25 hores per assolir els objectius temporals plantejats al diagrama de Gantt.

### 6.3.2 Recursos materials

A continuació, es descriuen els recursos materials que s'empraran durant el desenvolupament del projecte:

Recurs	Tipus	Finalitat
Ordinador	Eina de desenvolupament	Per escriure codi, testejar i redactar la memòria.
Git i Github	Eina de gestió	Per tenir un control de versions del projecte.
Gmail	Eina de comunicació	Per mantenir-me en contacte amb el director i el ponent
Ganttter	Eina de gestió	Per planificar la durada de cada tasca.
Skype Empresarial	Eina de comunicació	Per resoldre dubtes i validar tasques amb el director.
Truffle Framework	Eina de desenvolupament	Per desenvolupar i testejar els <i>Smart Contracts</i> .
VS Code	Eina de desenvolupament	Per desenvolupar i testejar.
Source Tree	Eina de gestió	Per gestionar repositoris Git.

## 7. Descripció de tasques

Aquestes són les tasques en les quals es divideix el projecte:

### 1) **Gestió i definició del projecte**

Aquesta és la fase de descripció i organització del projecte. Consisteix a deixar ben definit i especificat, mitjançant documents com aquest, l'abast, la planificació, els costos i qualsevol informació necessària. A més, també s'hauran de cercar els objectius que es volen assolir durant el desenvolupament del projecte i tota la informació possible que pugui ser útil pel projecte, per exemple, tecnologies que s'utilitzaran i l'estat de l'art.

### 2) **Implementació de la plataforma**

Aquesta fase conté tant l'aprenentatge de les tecnologies que s'utilitzaran pel desenvolupament de la plataforma web, com l'anàlisi de requisits, el disseny i la posterior implementació de la plataforma.

### 3) **Implementació de la Blockchain**

Aquesta fase conté l'aprenentatge del llenguatge Solidity, a més de l'anàlisi de requisits, el disseny i la implementació de la plataforma.

### 4) **Validació**

En aquesta fase es dissenyaran i realitzaran les proves per comprovar que l'aplicació funciona correctament.

### 5) **Documentació i presentació**

Finalment, l'última fase del projecte consisteix a redactar la memòria amb els resultats i preparar la defensa oral davant del tribunal que valorarà el treball realitzat.

## 8. Calendari

### 8.1 Estimació d'hores

<b>1. Gestió i definició del projecte</b>	<b>55h</b>
1.1 Definició de l'abast	20h
1.2 Planificació temporal	10h
1.3 Gestió econòmica i sostenible	10h
1.4 Presentació oral i document final	15h
<b>2. Implementació de la plataforma</b>	<b>250h</b>
2.1 Aprenentatge d'Angular	50h
2.2 Anàlisis de requisits	20h
2.3 Disseny	40h
2.4 Implementació	140h
<b>3. Implementació de la Blockchain</b>	<b>250h</b>
3.1 Aprenentatge de Solidity	50h
3.2 Anàlisis de requisits	20h
3.3 Disseny	40h
3.4 Implementació	140h
<b>4. Validació</b>	<b>60h</b>
4.1 Creació de proves	20h
4.2 Realització de tests	40h
<b>5. Documentació i presentació</b>	<b>120h</b>
5.1 Redacció de la memòria	90h
5.2 Preparació de la defensa	30h
<b>Total</b>	<b>735h</b>

## 9. Gestió econòmica del projecte

### 9.1 Introducció

En aquest apartat s'analitzarà el cost econòmic del projecte. Per això tindrem en compte diferents aspectes. El primer de tots és que no es consideraran costos els relatius al volum de producció, ja que no estem produint cap número específic d'unitats, sinó desenvolupant un servei software que s'oferirà als clients. Per tant, tots els costos seran considerats com costos d'estructura.

Per realitzar la gestió de costos es considerarà la següent classificació de costos: recursos humans, recursos materials (hardware i software) i costos generals. A més, també hem de comptar amb els costos de contingència i els imprevistos. Per cada un dels costos desglossats dins d'aquesta classificació s'especificarà si es tracta d'un cost directe o indirecte.

### 9.2 Estimació de costos

#### 9.2.1 Costos de recursos humans

Els costos de recursos humans són senzills de calcular, ja que només es disposa d'un treballador en pràctiques que s'encarrega de portar totes les tasques del projecte. Aquest treballador en pràctiques farà les funcions de cap de projecte, analista, dissenyador, desenvolupador i enginyer QA<sup>5</sup>.

Rol	Hores	Salari (€/h)	Cost
Cap de projecte	175 h	20 €/h	3.500 €
Analista	40 h	15 €/h	600 €
Dissenyador/a	80 h	12 €/h	960 €
Desenvolupador/a	380 h	12 €/h	4560 €
Enginyer/a QA	60h	12 €/h	720 €
<b>Total</b>	<b>735 h</b>		<b>10.340 €</b>

---

<sup>5</sup> Quality Assurance

Activitat	Hores	Recursos	Salari (€/h)	Cost (€)
<b>Gestió i inici del projecte</b>	55 h			
Definició de l'abast	20 h	Cap de projecte	20 €/h	400 €
Planificació temporal	10 h	Cap de projecte	20 €/h	200 €
Gestió econòmica i sostenible	10 h	Cap de projecte	20 €/h	200 €
Presentació oral i documentació final	15 h	Cap de projecte	20 €/h	300 €
<b>Implementació de la plataforma</b>	250h			
Aprenentatge d'Angular	50 h	Desenvolupador	12 €/h	600 €
Anàlisis de requisits	20 h	Analista	15 €/h	300 €
Disseny	40 h	Dissenyador	12 €/h	480 €
Implementació	140 h	Desenvolupador	12 €/h	1.680 €
<b>Implementació de la Blockchain</b>	250 h			
Aprenentatge de Solidity	50 h	Desenvolupador	12 €/h	600 €
Anàlisis de requisits	20 h	Analista	15 €/h	300 €
Disseny	40 h	Dissenyador	12 €/h	480 €
Implementació	140 h	Desenvolupador	12 €/h	1.680 €
<b>Validació</b>	60 h			
Creació de proves	20 h	Enginyer QA	12 €/h	240 €
Realització de tests	40 h	Enginyer QA	12 €/h	480 €
<b>Documentació i presentació</b>	120 h			
Redacció de la memòria	90 h	Cap de projecte	20 €/h	1.800 €
Preparació de la defensa	30 h	Cap de projecte	20 €/h	600 €



<b>Total</b>	735 h			10.340 €
--------------	-------	--	--	----------

Aquesta taula conté el cost directe per activitat en funció del rol que es requereix per dur-la a terme. També ens permet tenir una visió amb més detall d'on van a parar els diners del projecte en recursos humans i en quines tasques.

### 9.2.2 Costos de recursos materials

En aquest apartat s'inclouran els costos dels recursos materials de *hardware* i *software* del projecte.

Producte	Preu (€)	Unitats	Vida útil (anys)	Amortització/h	Hores	Amortització
Ordinador	600 €	1	4	0,075 €/h	430 h	32,25 €
Monitor Dell 24 "	280 €	1	4	0,035 €/h	430 h	15,05 €
<b>Total</b>	<b>880 €</b>					<b>47,30 €</b>

### 9.2.3 Costos generals indirectes

Dins dels costos generals, trobem tots aquells costos indirectes que no són ni de recursos humans, ni materials, com per exemple el preu del transport, de la llum, lloguer de l'edifici ...

Però com molts d'aquests costos els cobreix la empresa i a més per totes les persones que hi treballen, aquests són insignificants pel cost del projecte. Aquests costos generals són bàsicament el transport meu personal que es pot veure a la següent taula, juntament amb el cost estimat en euros (tenint en compte que la duració del projecte serà de 4 mesos, aproximadament).

Producte	Preu (€ / mes)	Cost (€)
T-Jove	30 €/mes	120 €
<b>Total</b>	<b>30 €/mes</b>	<b>120 €</b>

### 9.2.4 Contingència

Reservarem una quantitat de diners dins del pressupost per si apareix algun problema durant la realització del projecte.

Producte	Percentatge	Preu (€)	Cost (€)
Recursos humans	15 %	10.340 €	1551 €
Recursos materials	15 %	47,30 €	7,095 €
Recursos generals	15 %	120 €	18 €
<b>Total</b>		<b>10.507,3 €</b>	<b>1.576,095 €</b>

### 9.2.5 Imprevistos

En aquest apartat tenim en compte dos imprevistos contemplats durant la realització del projecte.

Producte	Probabilitat	Unitats	Preu	Cost (€)
Avaria ordinador	5 %	1	600 €	30 €
Retard implementació (15 dies)	20 %	75 hores	12 €/h	180 €
<b>Total</b>				<b>210 €</b>

### 9.2.6 Pressupost final

Per acabar fem un resum de totes les taules presentades anteriorment.

Concepte	Cost (€)
Recursos humans	10.340 €
Recursos materials	47,30 €
Costos generals	120 €

Contingència	1.576 €
Imprevistos	210 €
<b>Total</b>	<b>12.017,30 €</b>

### 9.3 Control de gestió

El control dels costos es realitzarà a través de reunions setmanals amb el director del projecte d'una durada aproximada de 30 minuts, on s'avaluarà el progrés de les diferents tasques.

Després de cada fase es realitzarà un informe resum que inclourà les hores emprades en cadascun dels rols i els recursos que s'han utilitzat. També es compararà el resultat de la fase amb el pressupost previ a la realització, i si aquesta variació és molt gran s'estudiarà com aconseguir que no torni a passar a les següents fases. L'índex de desviament serà del 15% tal com està establert a la contingència.

## 10. Sostenibilitat i compromís social

### 10.1 Ambiental

Aquesta solució tècnica pel projecte, basada en la tecnologia Blockchain, ajuda en l'aspecte ambiental ja que pel seu caràcter descentralitzat elimina els intermediaris i per tant, redueix la petjada ecològica generada per aquests.

### 10.2 Econòmic

Per considerar si un projecte és viable econòmicament és necessari realitzar un estudi de costos com el que s'ha fet en el primer apartat d'aquest document. Com es pot veure, la meitat dels costos del projecte aproximadament són de recursos humans. La resta de costos són costos inevitables que s'haurien de pagar fins i tot encara que no s'estigués desenvolupant cap mena de *software*, per tant el projecte és bastant viable econòmicament degut al seu baix cost de producció.

A més, molts dels recursos materials, al finalitzar el projecte, els hi quedarà una bona part de la seva vida útil i per tant, no caldrà invertir per nou material.

També aquest projecte, en utilitzar la Blockchain, permetrà estalviar costos econòmics d'intermediaris, que podrien ser altres empreses o persones, a través dels *Smart Contracts* i la seva manera de funcionar descentralitzada.

### 10.3 Social

L'aspecte social d'aquest projecte és el més important, sobretot per la comunitat de creadors de continguts. Actualment, aquests veuen com el seu contingut es viralitza sense recompensa econòmica pel creador o quan rep diners és perquè es fa a partir d'alguna distribuïdora. En canvi, d'aquesta manera com podran penjar el seu contingut a la plataforma, serà la pròpia Blockchain qui gestionarà els drets d'autor. En la compra d'un *asset*<sup>6</sup> es pagarà directament un percentatge del preu, definit als drets, a cadascun dels actors implicats (cantants, membres de la banda, estudio...).

Finalment, la solució proposada pertany a un projecte de codi obert que estarà públic per la comunitat de desenvolupadors per tal d'agafar idees i que podrà ser utilitzat com a base de futurs projectes.

---

<sup>6</sup> És un objecte que conté informació sobre propietat intel·lectual

## 10.4 Matriu de sostenibilitat

A la següent matriu de sostenibilitat es mostra la valoració final de la sostenibilitat del projecte.

	PPP	Vida Útil
Ambiental	Consum del disseny	Petjada Ecològica
	6	6
Econòmic	Factura	Pla de viabilitat
	7	8
Social	Impacte personal	Impacte Social
	7	8

## 11. Disseny

El projecte consisteix de 3 pilots de diferent contingut multimèdia:

- WebTV
- Music
- Photos

Em centraré a explicar principalment el pilot de WebTV i el de Music. Gran part dels components d'aquests dos pilots han estat implementacions meves. Tot i això, la demostració es realitzarà del pilot WebTV.

### 11.1 Pilot WebTV

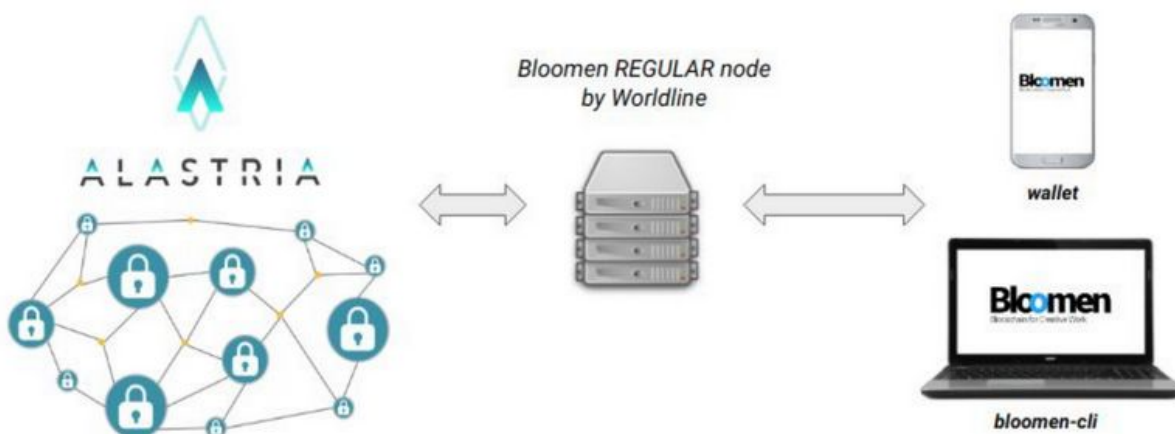
El pilot consisteix de 4 parts diferents que agreguen una solució completament operacional:

- **Bloomen Wallet / App:** Aplicació mòbil híbrida (instal·lable en Android i iOS) que funciona directament contra la Blockchain i realitza les tasques per firmar transaccions. La signatura de les transaccions al dispositiu ens permet estalviar-nos middlewares que posarien en dubte la seguretat a un sistema descentralitzat.
- **Bloomen Wallet / Smart contracts:** La lògica de negoci de l'aplicació ha estat desenvolupada en *Smart Contracts* desplegats a la Blockchain d'Alastria. Aquests *Smart Contracts* regulen el flux de les criptodivises i prevenen l'ús fraudulent dels serveis que els *Smart Contracts* ofereixen.
- **Bloomen Wallet / Bloomen-wallet-cli:** És una interfície de línia de comandes per interactuar amb la Blockchain.
- **Bloomen Wallet / Service demonstrator:** Es una demostració simple amb la qual es poden fer les accions/operacions bàsiques de l'usuari final. Més endavant entraré en detall d'aquesta solució.

L'aplicació ha estat construïda a sobre de Quorum<sup>[13]</sup>, què és un *fork* de la implementació d'Ethereum fet per JP Morgan. S'ha escollit fer-ho a sobre de Quorum perquè mentre que Ethereum és una xarxa pública i sense permisos, Quorum permet transaccions privades entre membres, i per tant és la millor opció per la idea del projecte.

Aquesta aplicació ha sigut desplegada en una xarxa de test hostejada per la Spanish business consortium (amb més de 200 companyies), Alastria, que té per objectiu crear un BC<sup>7</sup> business environment en el que negocis segurs puguin ser desenvolupats sota un entorn legal específic.

Com podem veure a la *Figura 5* el pilot funciona des d'un dispositiu mòbil (App), des d'un ordinador (Service Demonstrator o bloomen-wallet-cli) que interactua amb la Blockchain i els seus *Smart Contracts*.



*Figura 5. Descripció del prototip*

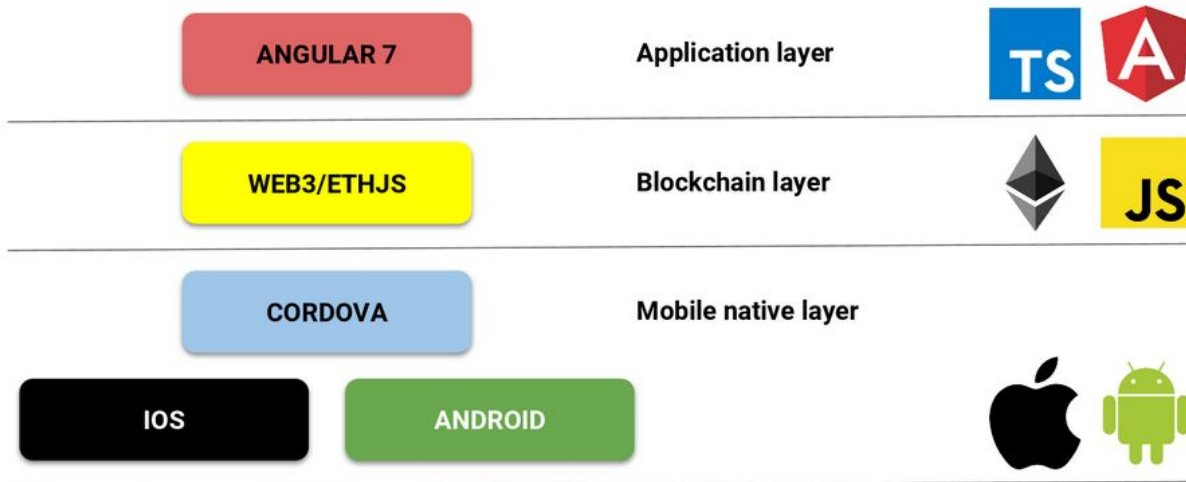
### 11.1.1 App

Aquesta aplicació per *smartphones* ofereix la possibilitat als usuaris finals d'interactuar amb les dades emmagatzemades a la Blockchain.

La idea d'oferir el servei al nombre més gran d'usuaris ens ha portat a utilitzar una tecnologia, Cordova<sup>[14]</sup>, que ens permet generar aplicacions per dispositius mòbils iOS i Android, web i *desktop*. Apache Cordova permet utilitzar les tecnologies estàndard per web com HTML5, CSS3 i JavaScript per fer un desenvolupament multiplataforma sense la necessitat d'utilitzar els llenguatges nadius de cada plataforma.

<sup>7</sup> Blockchain

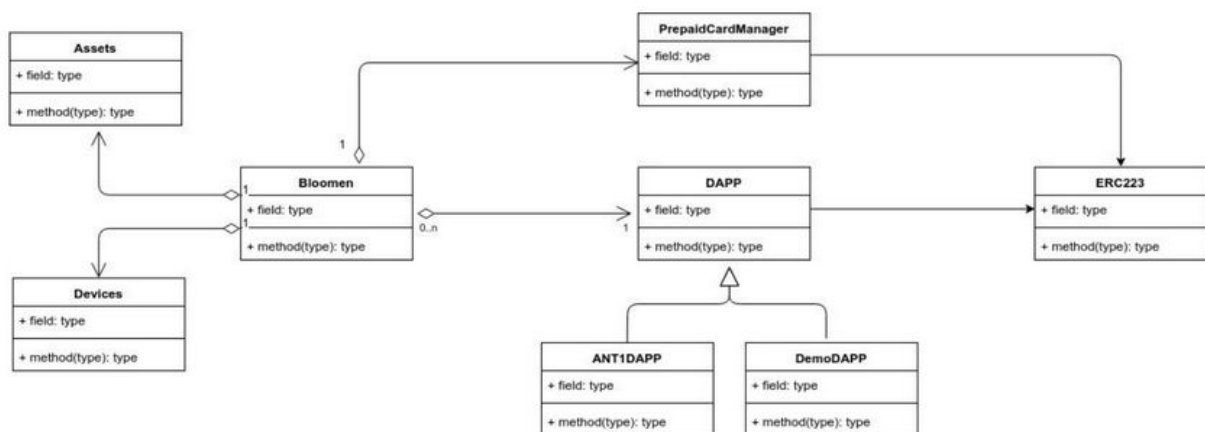
L'arquitectura de l'aplicació la podem veure a la *Figura 6* on observem des d'un punt de vista conceptual els diferents elements tècnics emprats en el desenvolupament.



*Figura 6. Arquitectura de l'aplicació*

### 11.1.2 Smart Contracts

Al següent diagrama podem veure les relacions entre els diferents *Smart Contracts*. *Bloomen*, és a dir l'aplicació, tindrà una llista de totes les *DApps* registrades, una llista d'*Assets*, una llista de *Devices*<sup>8</sup> i un *PrepaidCardManager* que gestionarà el *token* de l'aplicació.



*Figura 7. Diagrama dels Smart Contracts*

<sup>8</sup> Dispositius



Per tal d'oferir serveis de pagament des de l'aplicació de Bloomen cal un sistema d'adquisició de *tokens*. Per això, s'ha dissenyat un sistema basat en *prepaid cards*.

Podem observar un cas bàsic de distribució de targetes a la *Figura 8*. En aquest escenari tenim 3 usuaris diferents involucrats:

- **Service Provider**, entitat o persona propietària d'una DApp que ofereix diferents serveis a l'usuari final.
- **Service Broker**, venedor de targetes que té contacte directe amb els usuaris finals i és el primer que rep diners reals.
- **Service Consumer**, usuari final que vol consumir serveis que poden ser obtinguts a través de Bloomen.

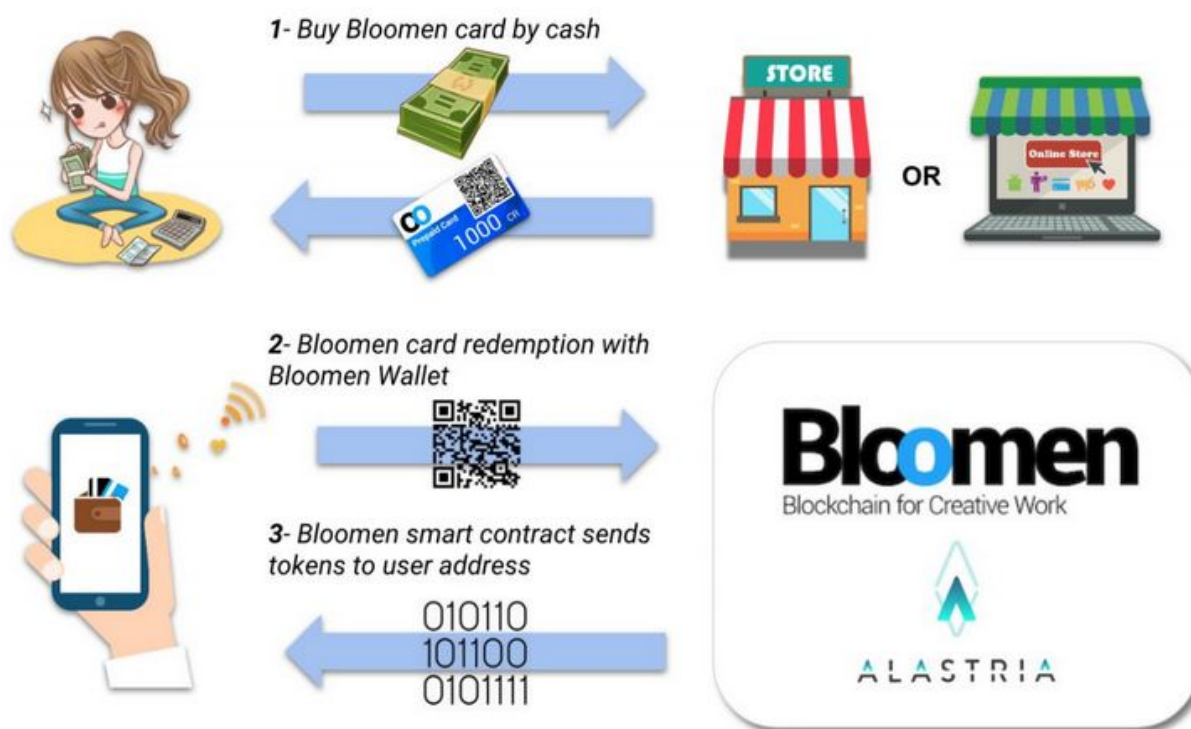


Figura 8. Cas bàsic de Prepaid Cards

### 11.1.3 Bloomen-wallet-cli

Aquesta senzilla eina ens permet assumir la identitat de diferents rols per tal d'enviar transaccions amb l'objectiu de validar el correcte funcionament del sistema.

Aquesta eina permet la creació de *prepaid cards* com a service provider, i gestionar les diferents DApps creades a la Blockchain. El service provider pot crear nous esquemes de pagament.

```
? Main -> Service Provider: (Use arrow keys)
> Menu Prepaid card
  Menu Demo
  Menu Schemes
  Menu DAPPS
  Back
  Exit
```

Figura 9. Menu Service Provider

Com a usuari, l'eina ens permet recuperar codis QR<sup>9</sup> amb la informació necessària que permet fer les proves requerides.

```
? Main -> Customer: (Use arrow keys)
> [C1] Generate Prepaid card QR
  [C2] Generate purchase operation QR
  [C3] Generate asset access request QR for device
  [SP13] dapp show QR
  Back
  Exit
```

Figura 10. Menu Service Costumer

### 11.1.4 Service Demonstrator

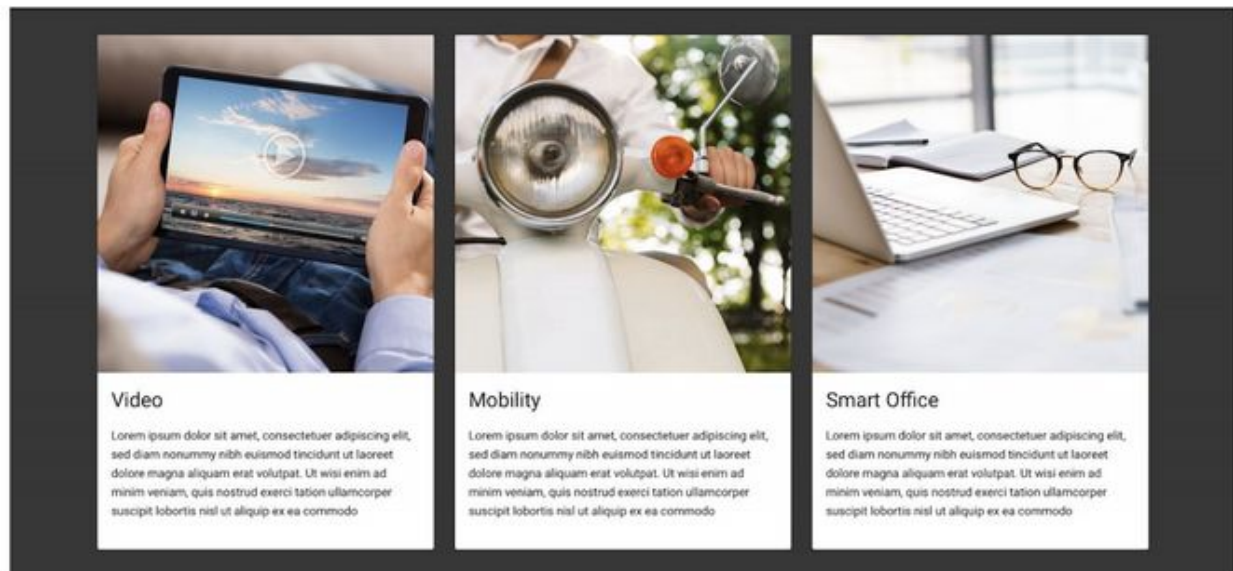
El pilot incorpora una altra demo, com a alternativa al bloomen-wallet-cli, que permet a l'usuari final fer servir les accions més importants, com comprar *assets* i autoritzar un dispositiu a reproduir un *asset*.

A continuació, mostro unes quantes captures de pantalla de les funcionalitats bàsiques:

---

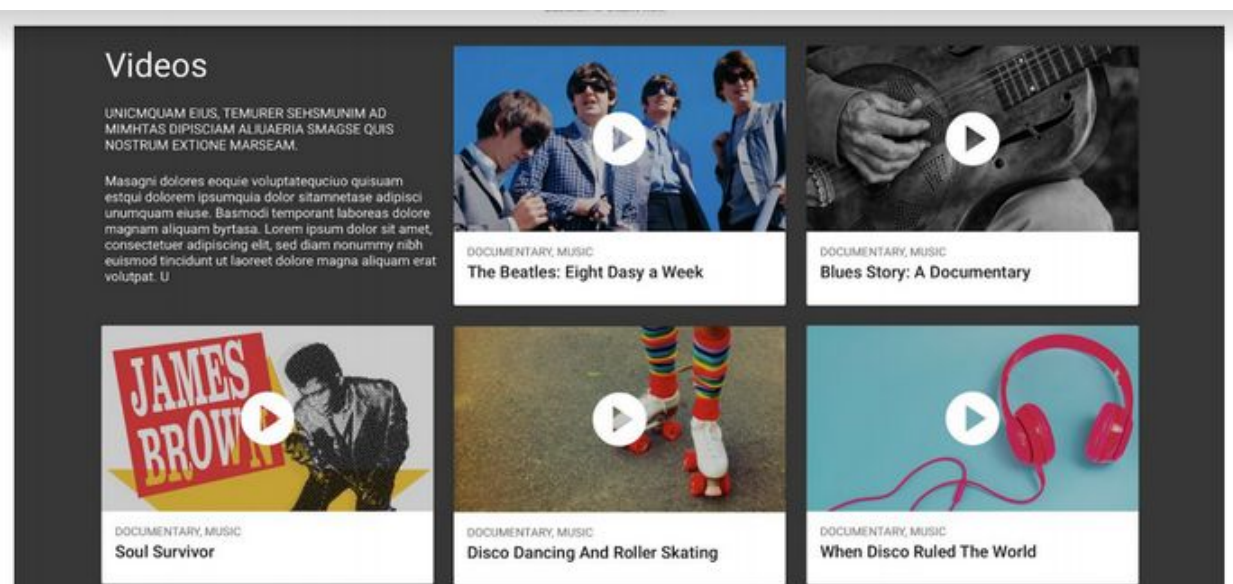
<sup>9</sup> Quick Response code

- Llista de demos, en funció del tipus de contingut:



*Figura 11. Llista de demos*

- Llista d'assets de l'apartat de *Vídeo* (Figura 12), *Mobility* (Figura 13) i *Smart Office* (Figura 14) :



*Figura 12. Llista de vídeos*

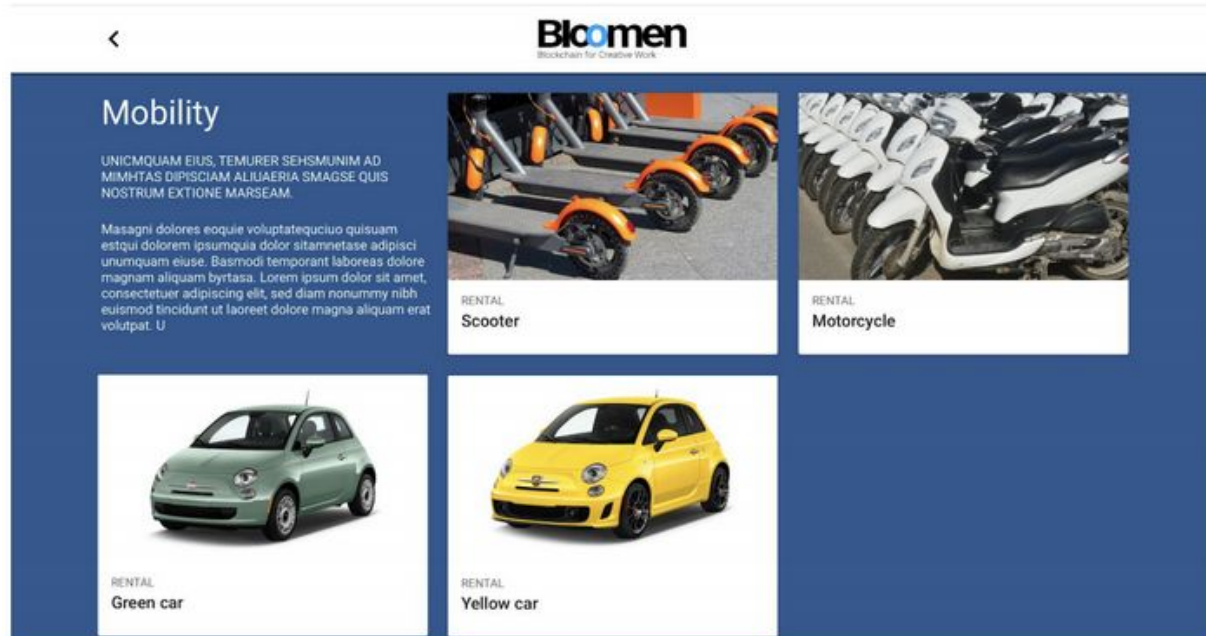


Figura 13. Llista de vehicles

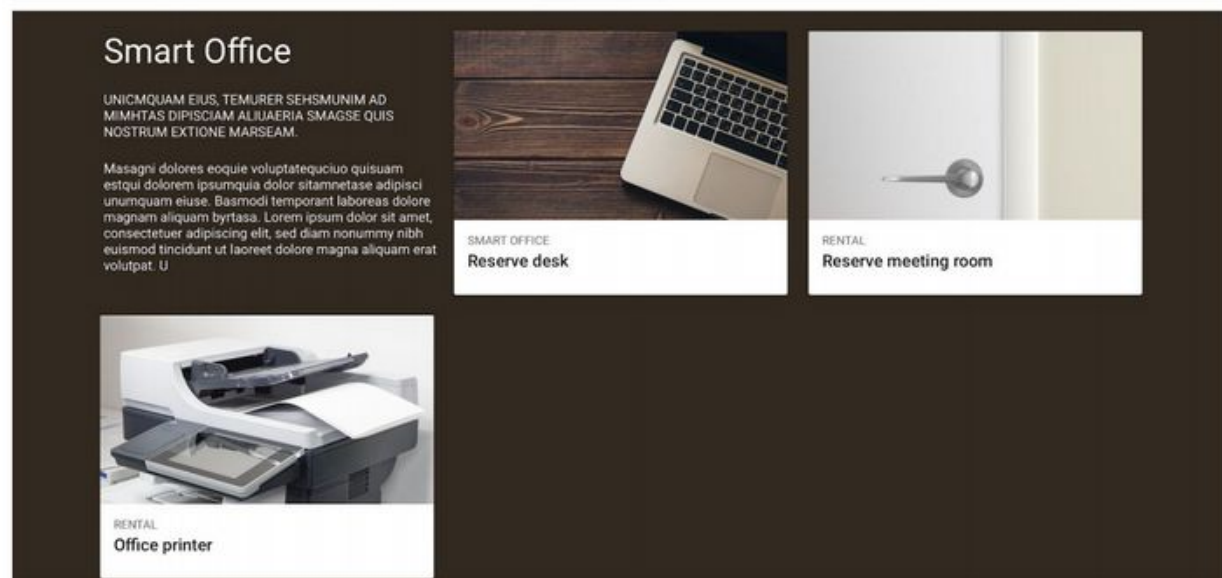


Figura 14. Llista d'articles per oficines



- Compra i reproducció d'un *asset* a través dels respectius codis QR:

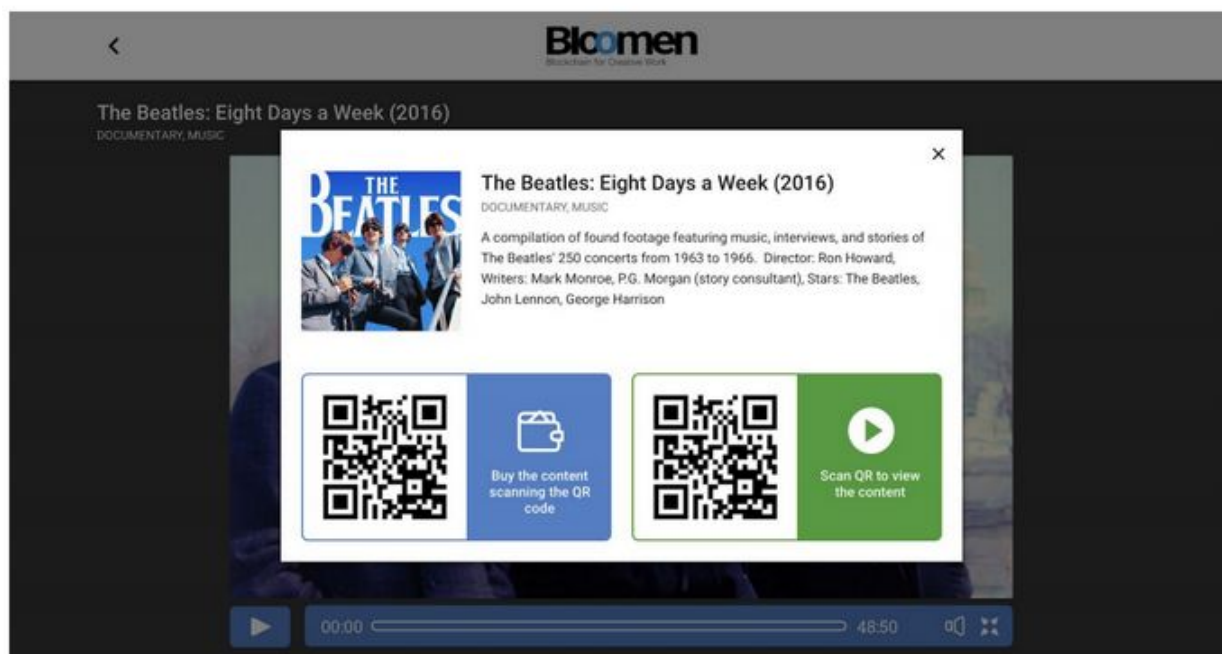


Figura 15. QRs per comprar i veure un vídeo

- Reproducció del vídeo després de la seva compra:

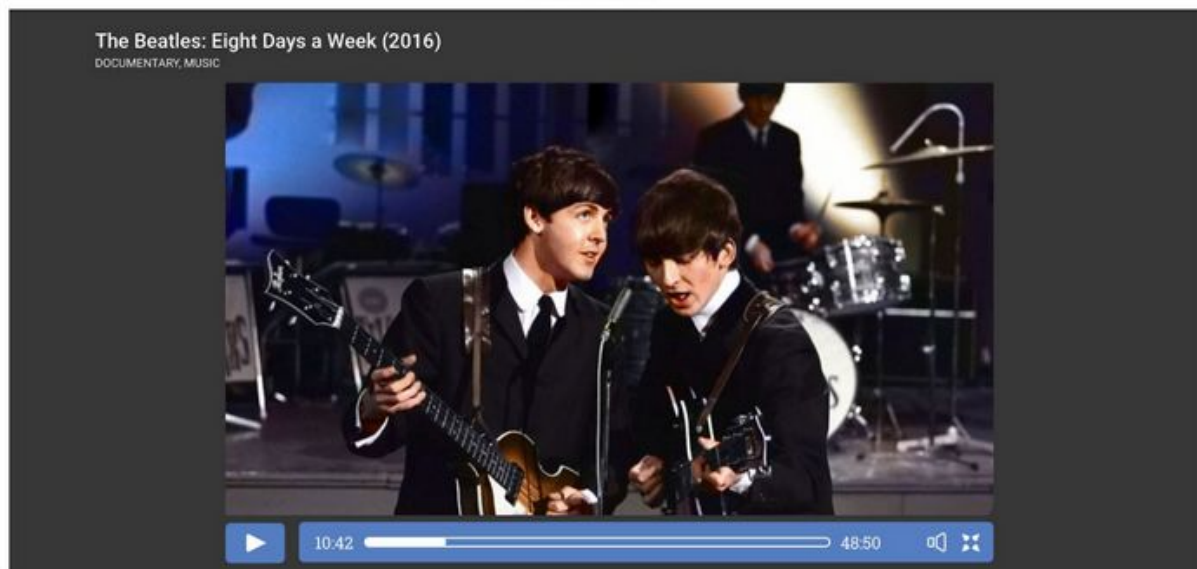
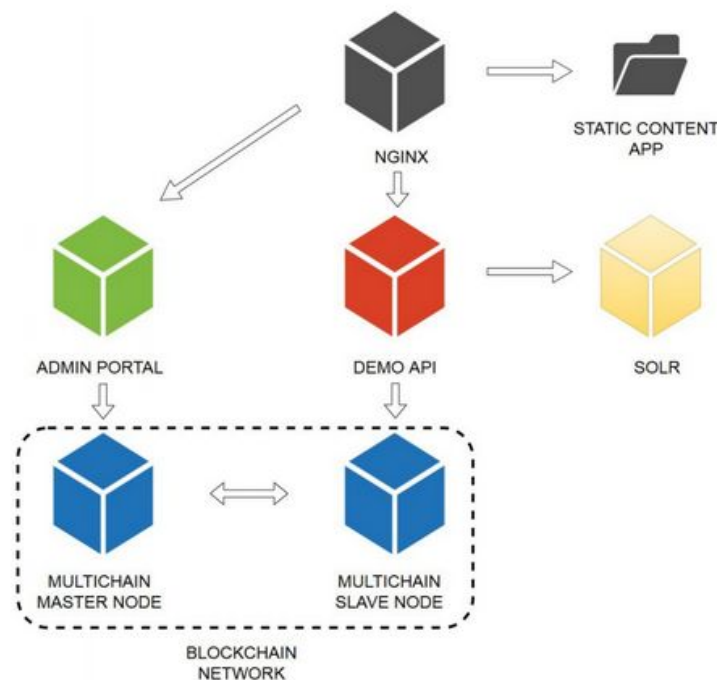


Figura 16. Reproducció d'un vídeo

## 11.2 Pilot Music

En aquest pilot inicialment s'exploren les capacitats de les Multichain per guardar metadades d'assets musicals amb l'objectiu de crear catàlegs compartits entre diferents entitats que comparteixen dades a través de la Blockchain.

També s'utilitzen tokens creats a la Blockchain per tal de dissenyar un *Claim* system.



*Figura 17. Arquitectura del Pilot de Música*

El pilot, inicialment, està construït utilitzant una xarxa Multichain<sup>[15]</sup> amb dos nodes (master i slave) i utilitza 4 subsistemes diferents (Admin portal, SOLR<sup>[16]</sup>, swagger “demo API” i demo app “static content”).

- Administration tool, el portal d'administració és un desenvolupament basat en un projecte existent de github<sup>10</sup> desenvolupat en PHP per la comunitat Multichain.

<sup>10</sup> <https://github.com/MultiChain/multichain-web-demo>

- SOLR, el search engine utilitza Apache SOLR software que permet crear cerques indexades per les dades guardades a la Blockchain.
- SWAGGER API, l'API REST que dóna accés a les funcionalitats del pilot ha estat desenvolupat en TypeScript.
- Demo Application, una aplicació que ha estat desenvolupada amb l'objectiu que l'usuari final no requereixi coneixement en Blockchain i pugui accedir a un gran nombre de funcionalitats. El desenvolupament de l'aplicació s'ha fet amb l'Angular ngx-Rocket application generator<sup>[17]</sup>.

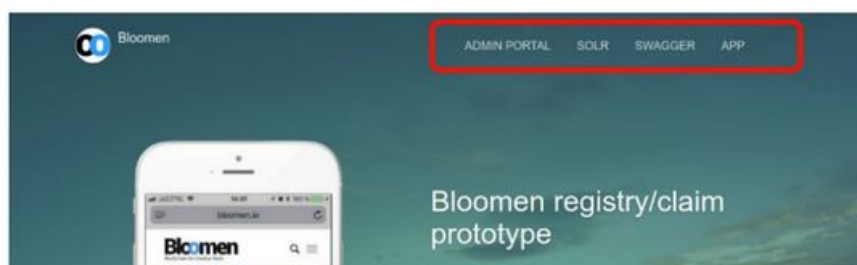


Figura 18. Pàgina inicial

Després d'explorar l'opció d'utilitzar la tecnologia Multichain pel Pilot de Music, es decideix canviar a Ethereum per aquest Pilot. Perquè tot i que Multichain ofereix una manera molt simple de crear xarxes Blockchain i té moltes eines molt potents per emmagatzematge i per crear tokens a partir de recursos, la manca de *Smart Contracts* i clients lleugers a Multichain ens va portar a descartar la plataforma Multichain sencera, i per tant escollir Ethereum.

Casos d'ús per l'aplicació:

Casos d'ús	Descripció
USMUL01	Com a usuari registrat vull crear noves dades multimèdia, per exemple <i>Bohemian Rhapsody by ACDC</i> .
USMUL02	Com a usuari creador vull buscar assets multimèdia, per exemple <i>Bohemian Rhapsody</i>

USMUL03	Com a usuari creador vull crear <i>claims</i> si algunes dades estan equivocades o hi ha qualsevol altre problema, per exemple <i>Queen claiming Bohemian Rhapsody's ownership</i> .
USMUL04	Com a usuari registrat vull rebre <i>claims</i> sobre els assets que tinc, per exemple <i>Queen claiming that ACDC is not Bohemian Rhapsody's composer</i> .
USMUL05	Com a usuari creador vull veure els meus <i>claims</i> , per exemple <i>registry user asking to Queen documentation about ownership in order to prove it</i> .



## 12. Desenvolupament

A continuació, explicaré cadascun dels components que he implementat. Aquests components formen part del Pilot de WebTV o/i del Pilot de Music.

### 12.1 REST API

A continuació explicaré una mica les parts de la REST<sup>11</sup> API<sup>12</sup> pel Pilot de Music.

La REST API està desenvolupada en NestJS<sup>[18]</sup>. S'aprofita el mòdul d'OpenAPI (SwaggerModule) que té NestJS per integrar l'api amb Swagger<sup>[19]</sup>. Podem veure a la següent figura com importo el mòdul a la línia 3 i també com a la línia 16 del codi generem la documentació de Swagger.

```
1  import { NestFactory } from '@nestjs/core';
2  import { AppModule } from './app.module';
3  import { SwaggerModule, DocumentBuilder } from '@nestjs/swagger';
4
5  require('dotenv').config();
6
7  async function bootstrap() {
8    const app = await NestFactory.create(AppModule);
9
10   const options = new DocumentBuilder()
11     .setTitle('NestJs Api')
12     .setDescription('The NestJs API description')
13     .setVersion('1.0')
14     .build();
15
16   const document = SwaggerModule.createDocument(app, options);
17   SwaggerModule.setup('api', app, document);
18
19   await app.listen(3000);
20 }
21 bootstrap();
```

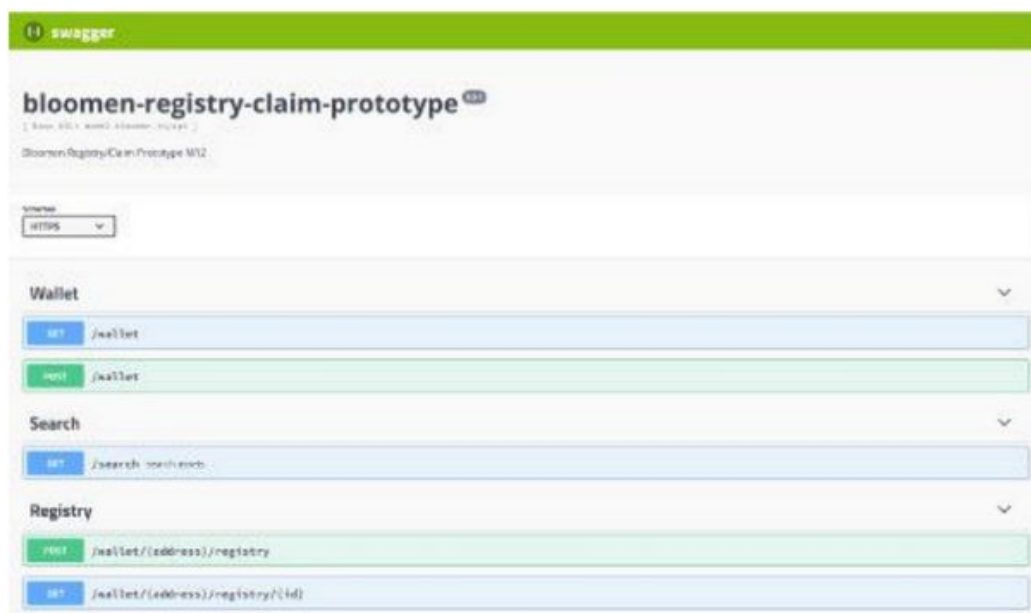
*Figura 19. main.ts de l'API*

<sup>11</sup> REpresentational State Transfer

<sup>12</sup> APplication Programming Interface

Swagger ens ofereix un portal per facilitar la interacció amb la REST API implementada. Aquesta REST API permet la interacció entre usuaris i els nodes de Blockchain d'una manera unificada.

A la següent figura podem veure el portal que genera Swagger per interactuar amb la REST API.



*Figura 20. Portal Swagger*

A la següent figura podem veure com amb `@ApiUseTags` indiquem a Swagger quin nom volem que tingui cada mètode de la REST API.

```
1  import { Get, Post, Controller } from '@nestjs/common';
2  import { WalletService } from './wallet.service';
3  import { Wallet } from './interfaces/wallet.interface';
4
5  //Swagger
6  import { ApiUseTags } from '@nestjs/swagger';
7
8  @Controller('/wallet')
9  export class WalletController {
10     constructor(private readonly walletService: WalletService) { }
11
12     @ApiUseTags('Wallet')
13     @Get()
14     async getWallet(): Promise<Wallet[]> {
15         return this.walletService.getWallet();
16     }
17
18     @ApiUseTags('Wallet')
19     @Post()
20     async postWallet(): Promise<Wallet[]> {
21         return this.walletService.postWallet();
22     }
23
24 }
```

Figura 21. Codi del /wallet

## 12.2 Smart Contracts

A continuació, explicaré les diferents eines utilitzades en la implementació dels *Smart Contracts*, aquests s'utilitzaran tant pel Pilot de WebTV com el de Music.

Aquests contractes han sigut desenvolupats utilitzant el *Truffle*<sup>[20]</sup> *development framework* per facilitar la creació de nous contractes si es requereix. Truffle té un seguit de funcionalitats que faciliten el desenvolupament dels Smart Contracts:

- Crear un projecte amb la següent estructura amb la comanda *truffle init*

- `contracts/` : Directory for [Solidity contracts](#)
- `migrations/` : Directory for [scriptable deployment files](#)
- `test/` : Directory for test files for [testing your application and contracts](#)
- `truffle-config.js` : Truffle [configuration file](#)

*Figura 22. Estructura projecte de Truffle*

- Compilar els *Smart Contracts*

```
truffle compile
```

*Figura 23. Comanda per compilar*

- Fer debug d'una transacció

```
$ truffle debug <transaction hash>
```

*Figura 24. Comanda per debuggar*

- Escriure tests en Solidity i executar-los

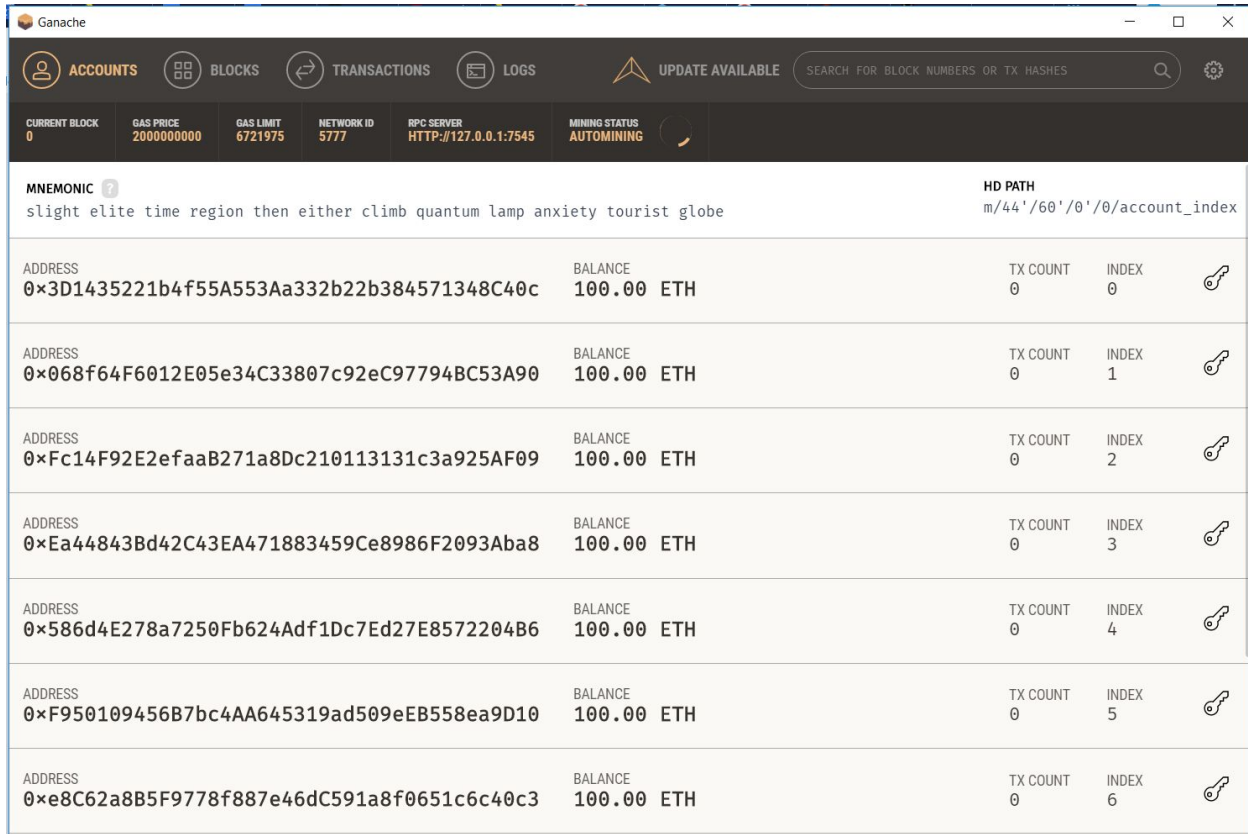
```
$ truffle test
Compiling ConvertLib.sol...
Compiling MetaCoin.sol...
Compiling truffle/Assert.sol
Compiling truffle/DeployedAddresses.sol
Compiling ../test/TestMetacoin.sol...

TestMetacoin
  ✓ testInitialBalanceUsingDeployedContract (61ms)
  ✓ testInitialBalanceWithNewMetaCoin (69ms)

2 passing (3s)
```

*Figura 25. Comanda per testear*

Una de les eines que utilitzo de Truffle és la Ganache, que permet crear de manera ràpida una Blockchain d'Ethereum local per llençar tests i executar comandes.



The screenshot shows the Ganache application window. At the top, there's a navigation bar with icons for ACCOUNTS, BLOCKS, TRANSACTIONS, and LOGS. Below this is a status bar showing various network metrics. The main area displays account information for a specific HD path. A table lists six accounts, each with an address, balance, transaction count, and index.

MNEMONIC		HD PATH
slight elite time region then either climb quantum lamp anxiety tourist globe		m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0x3D1435221b4f55A553Aa332b22b384571348C40c	100.00 ETH	0	0
0x068f64F6012E05e34C33807c92eC97794BC53A90	100.00 ETH	0	1
0xFc14F92E2efaaB271a8Dc210113131c3a925AF09	100.00 ETH	0	2
0xEa44843Bd42C43EA471883459Ce8986F2093Aba8	100.00 ETH	0	3
0x586d4E278a7250Fb624Adf1Dc7Ed27E8572204B6	100.00 ETH	0	4
0xF950109456B7bc4AA645319ad509eEB558ea9D10	100.00 ETH	0	5
0xe8C62a8B5F9778f887e46dC591a8f0651c6c40c3	100.00 ETH	0	6

*Figura 26. Interfície de Ganache*

Un dels punts més importants en el desenvolupament d'un *Smart Contract* és l'habilitat de limitar l'accés a certes funcionalitats d'acord a un rol específic, per tal de protegir el sistema. D'acord a aquesta premissa els contracts usen un *Ownership pattern* (openzeppelin-solidity) que protegeix la invocació de mètodes que poden canviar l'estat de la Blockchain.

El *NTUA Token* és la moneda de pagament del projecte. Les seves funcionalitats són útils per diferents casos d'ús de Bloomen. Per exemple, aquest *Smart Contract* pot ser instanciat i utilitzat de manera que un usuari pot veure la seva última transacció inclosa a la xarxa de Blockchain, la història de les transaccions, detectar una transacció passada, o fins i tot verificar si una quantitat ha sigut enviada a un altre usuari de manera satisfactòria.

Els *NTUA Token* segueixen totalment els ERC20/223 *tokens* i per tant, implementa un conjunt de funcions i esdeveniments.



```

1 // -----
2 // ERC Token Standard #20 Interface
3 // https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md
4 // -----
5 contract ERC20Interface {
6     function totalSupply() public view returns (uint);
7     function balanceOf(address tokenOwner) public view returns (uint balance);
8     function allowance(address tokenOwner, address spender) public view returns (uint remaining);
9     function transfer(address to, uint tokens) public returns (bool success);
10    function approve(address spender, uint tokens) public returns (bool success);
11    function transferFrom(address from, address to, uint tokens) public returns (bool success);
12
13    event Transfer(address indexed from, address indexed to, uint tokens);
14    event Approval(address indexed tokenOwner, address indexed spender, uint tokens);
15 }

```

Figura 27. Interfície ERC20

```

NtuaToken()
function () payable
function sellBack(uint256 amount, uint _price)
totalSupply()
balanceOf(owner)
approve(spender, value)
allowance(owner, spender, amount)
function transfer(address _from, address _to, uint _value) internal
function transfer(address _to, uint256 _value)
function managerTransferFrom(address _from, address _to, uint256
_value) returns (bool success)
Event Transfer(from, to, amount)
Event Approval(owner, spender, amount)

```

Figura 28. Ntua Token Smart Contract

Aquest contracte implementa la *fallback* function, una funció sense nom, marcada com a payable. L'objectiu de la funció és carregar els NTUA *tokens* per balancejar els comptes. Hi ha 3 funcions que implementen la transferència dels NTUA *tokens* entre comptes:

- i) la funció interna *\_transfer*, que transfereix una quantitat específica de NTUA tokens des d'un compte a un altre, si el balanç de l'usuari que envia té suficients estalvis. Si això es compleix, emet un esdeveniment *Transfer*, notificant aquells que estan seguint l'activitat de la Blockchain que la transacció ha sigut completada;
- ii) la funció *transfer* pot ser cridada des de fora d'un contracte. Crida a la funció *\_transfer* per tal d'enviar els tokens del sender de la transacció al receptor especificat.

iii) la funció *ManagerTransferFrom*. Aquesta funció es crida des del Asset Manager contract, que explicaré a continuació, i transfereix NTUA *tokens*, quan algú compra dades. Només funciona si és invocada pel contracte Asset Manager.

Web Functionalities Cryptocurrency/Micropayments	WebTV	Music
Purchase asset	X	
Transfer funds to address	X	
View balance of wallet	X	
View Transactions of user	X	

*Figura 29. Ntua Token funcionalitats*

El Smart Contract del Asset Manager s'encarrega de mantenir tots els creadors registrats, realitzant i guardant les transaccions que involucren els assets. El Asset Manager també manté tots els assets registrats. En aquest contracte s'han creat structs per tal de guardar tots els detalls (per exemple, price, tags, information) i l'única adreça del seu propietari. Això permet al creador d'un nou *asset* (per exemple, un article, foto, àudio, vídeo...) especificar els detalls del seu asset.

```
struct media_content {
    address owner;
    string URI;
    uint256 price;
    string tag;
    string info;
}
```

*Figura 30. Estructura media\_content*



```
event Uploading(
    address owner,
    string URI,
    uint256 price,
    string tag,
    string info
);
event Retrieving(address owner, string info);
```

*Figura 31. Esdeveniments Uploading i Retrieving*

Web Functionalities Browsing/Searching Assets	WebTV	Music
Upload asset	X	X
Edit asset	X	X
List/Filter/Search asset	X	X
View Asset	X	X

*Figura 32. Asset Manager funcionalitats*

El Copyright Smart Contract suporta diferents esquemes de pagament. Després de la compra d'algun *asset* (imatge, cançó...), un percentatge del pagament es transfereix a diferents entitats d'acord als drets definits (cantant, membres del grup, estudi de gravació...). També suporta diferents esquemes de facturació, per exemple, un usuari registrat i verificat podria seguir un altre esquema de pagament. Més específicament i també com a exemple una ONG<sup>13</sup> podria utilitzar assets sense pagar o a través d'un preu reduït. A més, l'smart contract controla la comunicació entre els diferents *Smart Contracts* (Asset Smart Contract i Token Smart Contract). Inclou diferents estructures de dades, mètodes i esdeveniments, alguns dels quals apareixen a la següent llista:

- Structs:
  - `media_ownership`: percentatge de la propietat
  - `media_content`: detalls sobre l'*asset*

<sup>13</sup> Organització No Governamental

- Functions:
  - createMedia: carregar els detalls d'un nou *asset* a la Blockchain
  - deleteMedia: eliminar un *asset* de l'estructura de dades del smart contract
  - updatePrice: actualitzar informació d'un *asset*
  - updateUrl: actualitzar informació d'un *asset*
  - buyMedia: útil per comprar un *asset*, es comunica amb altres *Smart Contracts*
- Events:
  - MediaCreated: emès a la xarxa Blockchain, quan algun contingut multimèdia és creat
  - MediaDeleted: emès a la xarxa Blockchain, quan un *asset* és eliminat
  - MediaChangedInfo: emès quan la informació d'un *asset* s'ha canviat

```
struct media_ownership {  
    address owner1;  
    uint percentage1;  
    address owner2;  
    uint percentage2;  
    ....}  
  
struct media_content {  
    string mediaTitle;  
    string artist;  
    uint year;  
    string url;  
    uint price;}
```

Figura 33. Estructura *media\_ownership* i *media\_content*

```
mapping(uint => media_ownership[]) mediaOwnershipTable;
mapping(uint => media_content[]) mediaContentTable;

function createMedia(
    address o1, uint p1, address o2,
    uint p2, string url,uint price,
    string mediaTitle, string artist, uint year) external {
    mediaOwnershipTable[idStable].push(media_ownership(o1, p1, o2,
    p2));
    mediaContentTable[idStable].push(media_content(mediaTitle,
    artist, year, url, price));
    emit MediaCreated(mediaTitle, artist, year, price, idStable);
    idStable++;
}
```

Figura 34. Mapping i createMedia

Web Functionalities Browsing/Searching Assets	WebTV	Music
%ownership, rights management	X	X

Figura 35. Copyright funcionalitat

El *PrepaidCardManager* emmagatzemarà un número de *tokens* a la Blockchain, que representaran les *prepaid cards*. La creació d'una targeta es fa mitjançant el mètode *addCard* que generarà una crida interna al mètode *mint* que generarà més *tokens* (i per tant, *prepaid cards*). Aquest mètode està restringit al propietari del contracte (ERC20). El mètode contrari és el *burn* que permet destruir *tokens* quan sigui necessari.

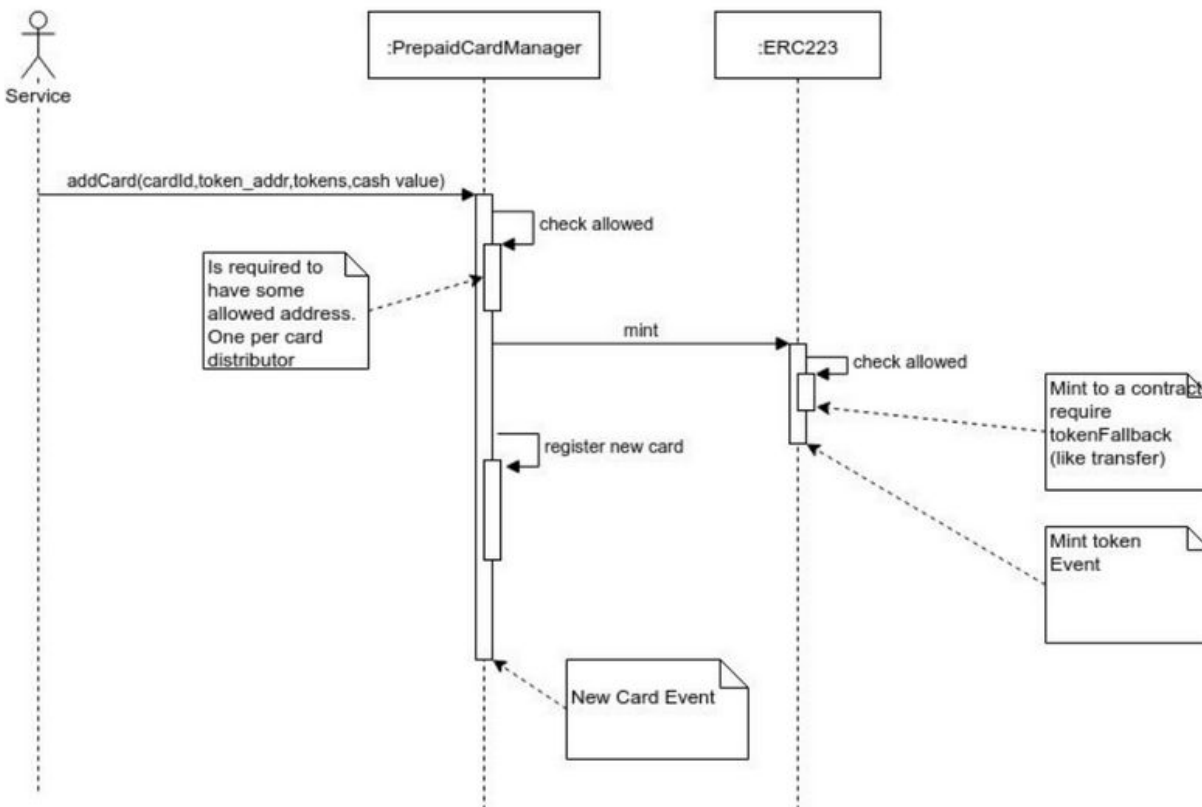


Figura 36. Diagrama de seqüència del `addCard`

A la *Figura 38* podem veure com un *Service Costumer* adquireix una tarjeta després de pagar amb diners reals al *Service Broker*. Un cop que la transacció és realitzada, modificarà la *prepaid card*, per tant podrà ser consumida. L'activació gravarà un esdeveniment que permetrà al *Service Provider* seguir les targetes i crear un flux de diners reals entre el *Customer* i el *Retailer*.

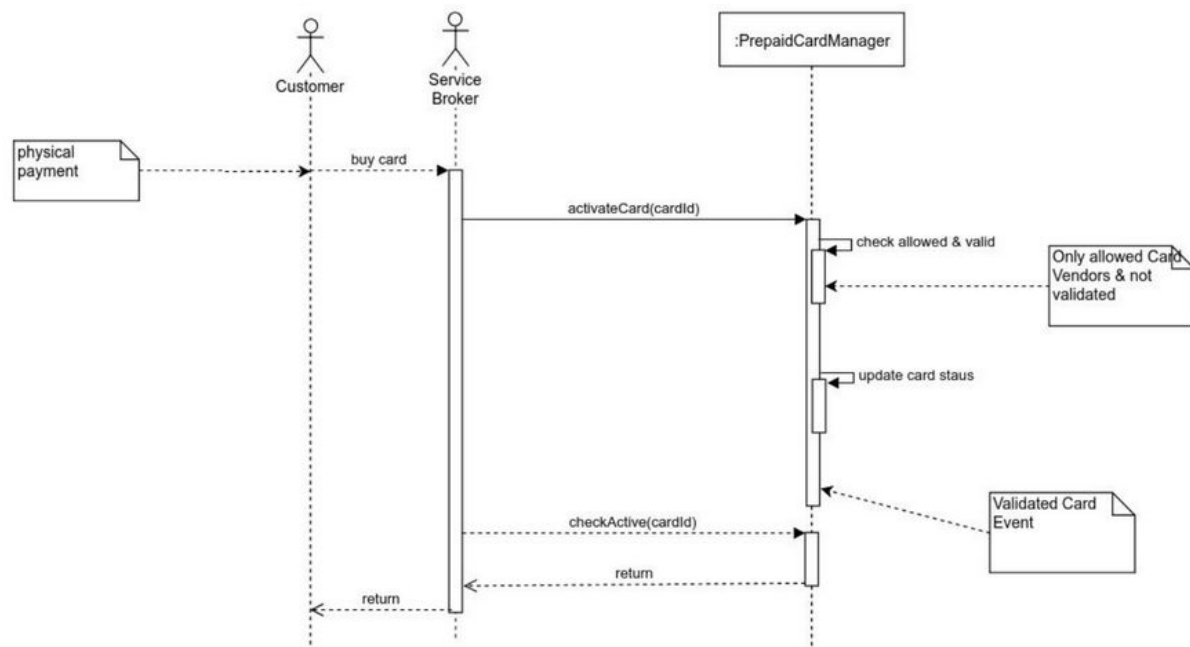


Figura 37. Diagrama de seqüència del buyCard

Per acabar el cicle de vida de la *prepaid card*, el end user reclamarà els *tokens* de la *prepaid card*. Aquesta acció elimina la *prepaid card* de l'inventari i transfereix els punts del contracte al compte de l'usuari.

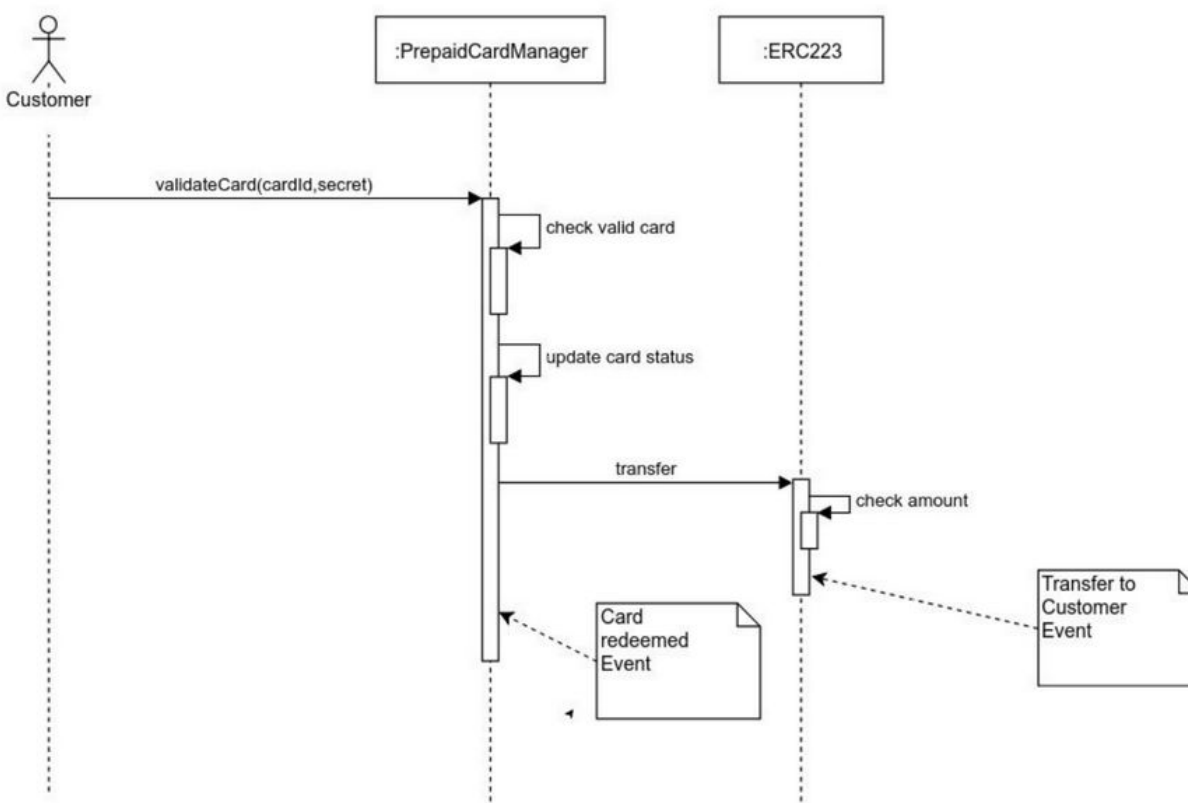


Figura 38. Diagrama de seqüència de `validateCard`

## 12.3 Interfície

La Demo Application és una petita aplicació, que s'utilitzarà dins del Pilot de Music, construïda amb l'objectiu de fer més fàcil la interacció pels usuaris finals. Utilitza la REST API per interactuar amb un node específic i les seves funcionalitats són *wallet management*, *new data registration* i *Claim creation* a la Blockchain.

Aquesta aplicació ha sigut desenvolupada amb l'objectiu que l'usuari final no requereixi coneixements sobre Blockchain i permeti provar gran part de les funcionalitats.

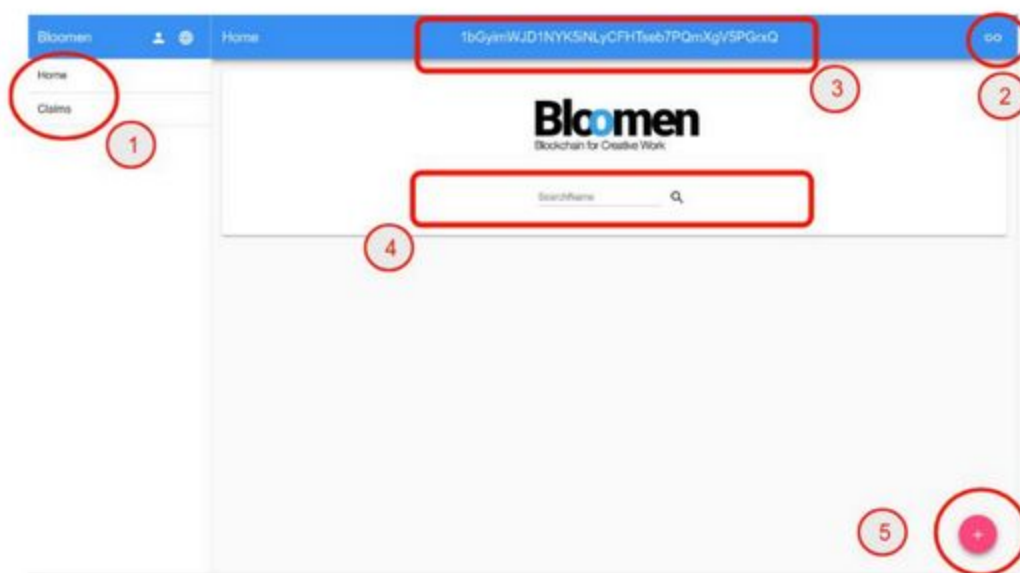


Figura 39. Pantalla inicial

L'aplicació cobreix unes funcionalitats bàsiques que són les següents:

1. Accedir a les funcionalitats del *Home* i del *Claim*. La secció de *Home* ens permet cercar i crear *assets* musicals, i la secció *Claim* permet gestionar les accions relacionades amb els *claims*.
2. Desplegable per la creació i selecció d'adreces.
3. Veure quina adreça s'està utilitzant.
4. Funció de cerca de dades guardades a la Blockchain.
5. Crear nous *assets* musicals.

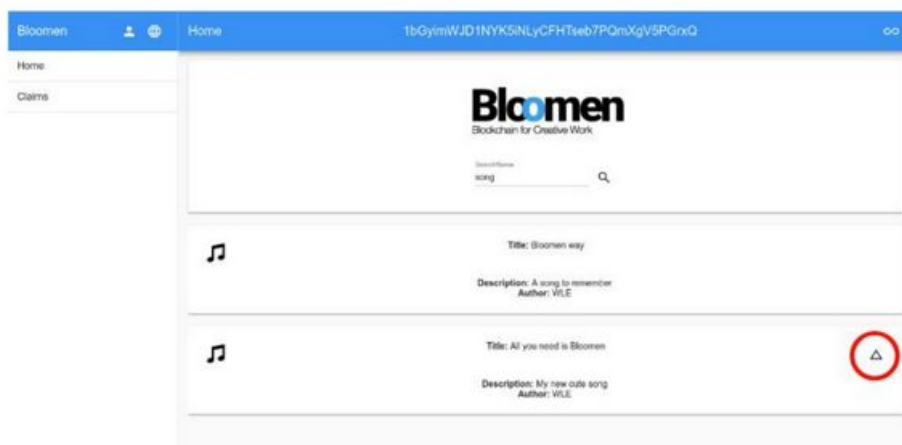


Figura 40. Resultat d'una cerca

El resultat de la cerca ens permet fer un *claim* a les cançons que no ens pertanyin. El *claim* consisteix només en un sistema de missatgeria que es guarda a la Blockchain.

La secció de *claim* ens permetrà consultar els *claims* dels *assets* de les quals en som propietaris tenint en compte l'adreça seleccionada (header).

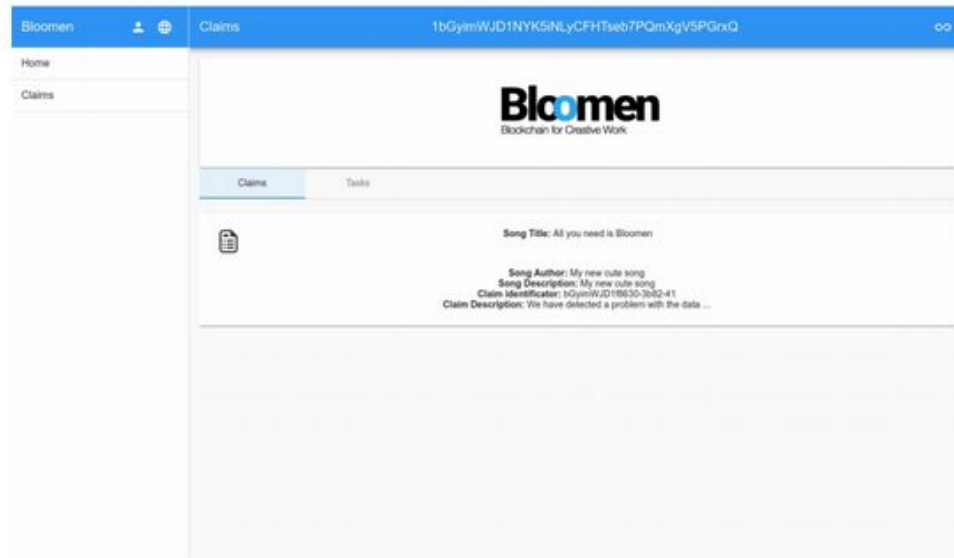


Figura 41. Claims tab

La secció de tasques ens ensenya els claims pendents que tenim assignats a l'adreça actualment seleccionada.



Figura 42. Tasks tab



La icona ressaltada ens permet contestar a un *claim* i generar una nova tasca que s'enviarà a l'adreça originària del *claim*.

### USMUL01 - As a registry user I want to create new media data

1. Premem el següent botó que està localitzat a la part baixa i a la dreta de la pantalla.



*Figura 43. Botó per afegir nous assets*

2. Omplim el formulari amb les dades de la nova cançó que volem publicar.



The image shows a mobile app interface for 'Bloomen Blockchain for Creative Work'. A modal form titled 'Add Registry' is displayed in the center. It contains three input fields: 'Name', 'Author', and 'Description'. At the bottom of the form are two buttons: 'Close' and 'Submit'.

*Figura 44. Formulari per afegir un nou asset*

3. Premem el botó de *Submit* i les dades es guardaran a la Blockchain associades a l'adreça seleccionada al *header* de l'app.



*Figura 45. Adreça seleccionada*

Cada adreça representa una identitat diferent respecte a la Blockchain i, en aquest cas, estableix el propietari de les dades que ha publicat el nou contingut.

Si volem tenir contingut creat per alguna adreça més, només hem de seleccionar una adreça diferent i repetir l'acció de crear una nova cançó.

Després d'uns segons, s'obté la confirmació de la transacció i la indexació dels continguts, podem accedir a aquestes dades des del cas d'ús de la cerca (USMUL02).

### **USMUL02 - As a creator user I want to search media assets**

1. Escriure el string que es vol cercar al camp i fer click a la icona de la lupa.



*Figura 46. Cerca d'una cançó*

En aquest cas d'ús, els *claims* han sigut limitats a cançons que no pertanyen a l'adreça que s'està utilitzant.

### USMUL03 - As a creator user I want to create *claims* if data is mistaken or has any other issues

1. Començant pel resultat de la pantalla de cerca (USMUL02), podem crear un *claim* per cançons que no són pròpies i enviar aquest *claim* al propietari automàticament.



*Figura 47. Crear un claim d'una cançó*

2. Omplim la descripció del *claim* que volem enviar al propietari de les dades i farem clic al botó de *Submit*.

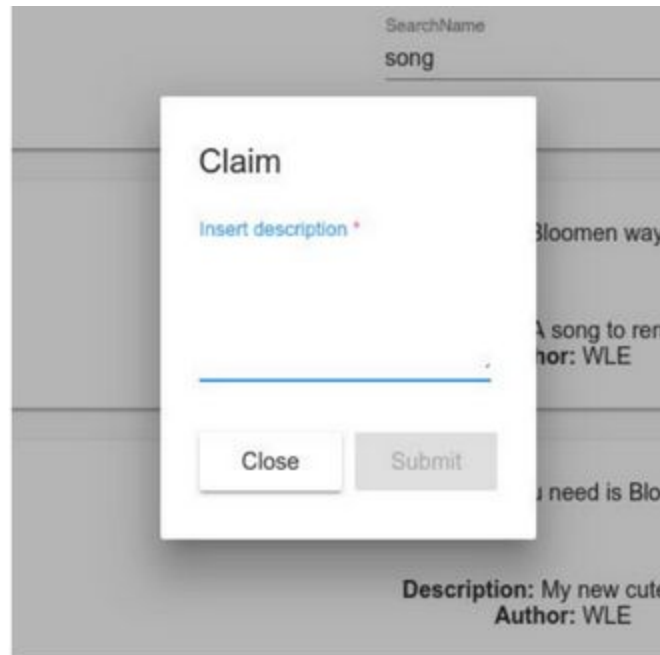


Figura 48. Formulari descripció del claim

**USMUL04 - As a registry user I want to receive *claims* about assets that I maintain**

1. Seleccionarem la *task tab* on trobarem tots els claims pendents de resposta assignats a l'adreça seleccionada (header), com podem veure a la Figura 43.
2. Si volem contestar el *claim*, generarem una nova tasca que apareixerà a la llista de tasques de l'originador del *claim*.

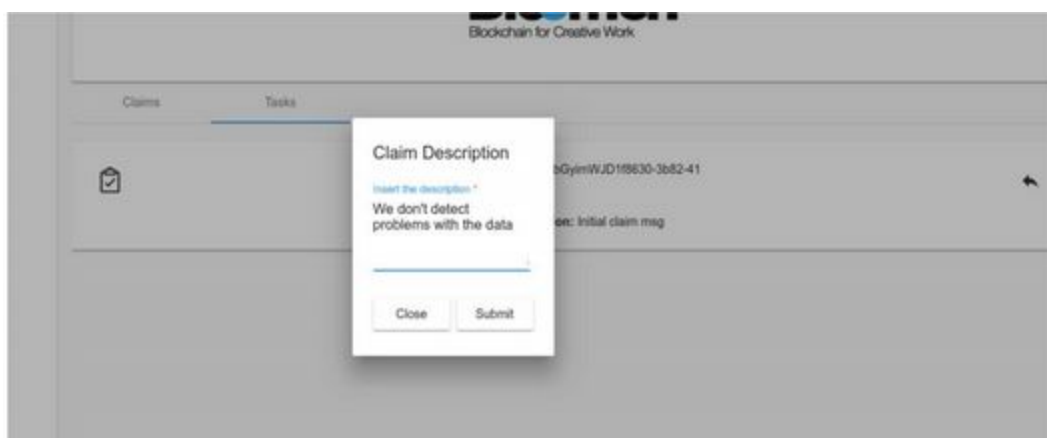


Figura 49. Descripció del claim

3. Si canviem l'adreça que havia originat el *claim*, podem veure com ara té una tasca pendent amb la informació afegida en el pas anterior.



*Figura 50. Tasks tab amb nova task*

Totes les dades i missatges enviats entre adreces són persistents a la Blockchain. Aquestes dades poden ser llegides per qualsevol altre node que pertany a la xarxa i permet tenir un seguiment complet dels canvis d'estat en la informació relacionada amb la cançó.

#### **USMUL05 - As a creator user I see my claims**

Com a propietari d'un *claim* puc accedir sempre a les seves dades des del *claims tab*.



Figura 51. Claim tab amb nou claim

## 12.4 Llibreria NPM

Aquesta llibreria<sup>[21]</sup> s'utilitza en el bloomen-wallet-cli del pilot de WebTV. Aquesta s'utilitza per a fer més fàcil treballar en format Json i JsonPath. Consta de 3 funcionalitats principals:

1. De Json a JsonPath
2. De JsonPath a Json
3. Comparar 2 Json's

A la llibreria s'utilitzarà una classe, JsonPathPair, que representarà el format JsonPath.

El constructor d'un JsonPathPair és el següent:

```
public constructor(private path: string, private value: any, private type: string, private diff: string);
```

Figura 52. Constructor

A la següent figura es pot veure com es crearia un objecte amb format JsonPathPair a la llibreria.

```
/*{
  "a": "a",
  "b" : {
    "c" : 1
  }
}*/

let JsonPathPairs: JsonPathPair[] = [];
JsonPathPairs.push(new JsonPathPair("a", "a", "string", ''));
JsonPathPairs.push(new JsonPathPair("b.c", 1, "number", ''));
```

*Figura 53. Creació d'un JsonPathPair*

PATH	VALUE	TYPE
a	"a"	string
b.c	1	number

*Figura 54. Valor del JsonPathPair*

La primera funcionalitat és la de transformar un objecte Json en un JsonPath. La funció de la llibreria que implementa aquesta funcionalitat es diu *marshall*.

```
public marshall(obj: any, path: string, JsonPathPairs: JsonPathPair[]): JsonPathPair[]
```

*Figura 55. Mètode marshall*

En el primer paràmetre inserim el json. En el segon paràmetre i en el tercer inserirem sempre el mateix valor, "" i [] respectivament. En la següent figura podem veure com s'utilitza.

```
let json = {  
  "a": "a",  
  "b": {  
    "c": {  
      "d": "d",  
      "e": "e"  
    },  
    "f": "f"  
  }  
}  
  
const marshalled = library.marshall(json, "", []);
```

Figura 56. Transformar Json en un JsonPath

PATH	VALUE	TYPE
a	"a"	string
b.c.d	"d"	string
b.c.e	"e"	string
b.f	"f"	string

Figura 57. Valor del JsonPath

La segona funcionalitat, que transforma un JsonPath en un objecte Json, està implementada a la funció *unMarshall*.

```
public unMarshall(JsonPathPairs: JsonPathPair[]): any
```

Figura 58. Mètode unMarshall

El primer paràmetre serà el nostre JsonPath que volem transformar en un objecte Json. A les següents figures podem veure com s'utilitza aquesta funció.



PATH	VALUE	TYPE
a	"a"	string
b.c.d	"d"	string
b.c.e	"e"	string
b.f	"f"	string

Figura 59. Valor del JsonPath

```
const unmarshalled = library.unMarshall(marshalled, "", []);

/*unmarshalled = {
  "a": "a",
  "b": {
    "c": {
      "d": "d",
      "e": "e"
    },
    "f": "f"
  }
}*/
```

Figura 60. Transformar JsonPath en un Json

La tercera funcionalitat ens permet comparar 2 objectes Json i el mètode, que es diu *compareJsonPath*, retornarà un objecte *JsonPathPair[]*. A cada element apareixerà la diferència entre els dos Json (Added, Modified o Deleted) i el valor actual. A les següents figures podem veure un exemple de com funcionaria.

```
public compareJsonPath(before: any, after: any): JsonPathPair[]
```

Figura 61. Mètode *compareJsonPath*

```
let before = {
  "a": "a",
  "b": {
    "c": {
      "d": "d",
      "e": "e"
    },
    "f": "f"
  }
}

let after = {
  "a": true,
  "b": {
    "c": {
      "d": "d",
      "g": "g"
    },
    "f": 2
  }
}

const compared = library.compareJsonPath(before, after);
```

*Figura 62. Comparar 2 Json's*

PATH	VALUE	TYPE	DIFF
a	true	boolean	Modified
b.c.e	"e"	string	Deleted
b.c.g	"g"	string	Added
b.f	2	number	Modified

*Figura 63. Valor de la comparació*

## 13. Conclusions

L'objectiu principal d'aquest Treball de Final de Grau ha estat explorar les diferents tecnologies Blockchain per tal de generar un model que doni resposta als reptes actuals de l'àmbit del contingut multimèdia, com per exemple gestionar els drets d'autor, comprar contingut, ... entre d'altres.

Des del punt de vista tecnològic, he après a programar en NestJS, Solidity i Angular. Aquestes tecnologies són bastant recents, però cada cop més projectes i desenvolupadors les utilitzen. També he conegut com funciona la tecnologia Blockchain conceptualment i he conegut les diferents opcions dins d'aquesta tecnologia, com Ethereum, Multichain o Quorum. A més d'aprendre noves tecnologies, també he utilitzat les tecnologies web HTML5, CSS3 i JS.

Gràcies a les diferents assignatures de programació cursades des de l'inici de la titulació, he sigut capaç d'implementar un codi reutilitzable, eficient, net i que es pugui mantenir. A l'assignatura de PTI se'ns va introduir la tecnologia Blockchain i la interfície API, gràcies a això ha estat més senzill entendre els nous llenguatges. Les assignatures de PI, TXC i SI m'han ajudat a ser capaç d'entendre, avaluar i gestionar serveis web tenint en compte la seguretat d'aquests. La programació web està plena d'esdeveniments síncrons i asíncrons, conceptes que a SOA es van arribar a explicar i implementar internament.

Des del punt de vista de la gestió del projecte, he après a planificar, de manera acurada, la feina a fer durant la realització del TFG.

Després de finalitzar el projecte, s'han complert els objectius plantejats inicialment sense desviacions importants, s'ha desenvolupat una plataforma per crear, compartir i utilitzar contingut multimèdia i s'ha desenvolupat una sèrie de *Smart Contracts* per emmagatzemar i gestionar el contingut. A més, s'ha aconseguit desenvolupar una aplicació operativa i funcional. Per tant, certificant que la tecnologia Blockchain és una molt bona opció dins de l'àmbit de contingut multimèdia.

## 14. Apèndixs

### 14.1 Gantt

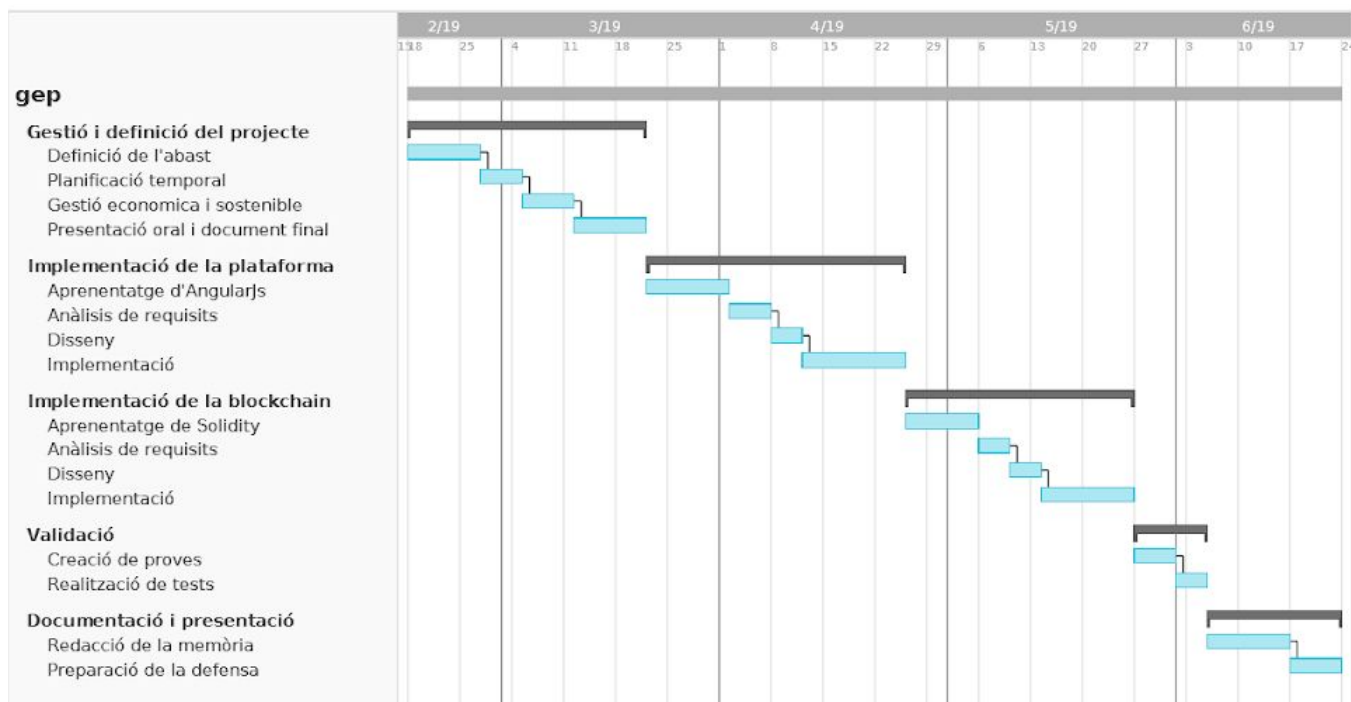


Figura 64. Diagrama de Gantt

### 14.2 Repositoris

- Repositori del projecte

<https://github.com/bloomenio>

- Repositori de la llibreria NPM

<https://github.com/worldline-spain/json-path-value>

## 15. Bibliografia

[1] Home - Bloomen [online]

Disponible a: <http://bloomen.io/> [Febrer, 2019]

[2] What is the Blockchain? Explaining the Tech Behind Cryptocurrencies - The New York Times [online]

Disponible a:

<https://www.nytimes.com/2018/06/27/business/dealbook/blockchains-guide-information.html> [Febrer, 2019]

[3] Qué son los 'smart contracts ' o contratos inteligentes | BBVA

Disponible a:

<https://www.bbva.com/es/smart-contracts-los-contratos-basados-blockchain-no-necesitan-abogados/> [Febrer, 2019]

[4] Hubii [online]

Disponible a:

<https://www.hubii.com/> [Febrer, 2019]

[5] Steemit [online]

Disponible a: <https://steemit.com/> [Febrer, 2019]

[6] Steemit [online] Simple decentralized app architecture

Disponible a: <https://steemit.com/ipfs/@admiboss/simple-decentralized-app-architecture> [Març, 2019]

[7] Ethereum [online]

Disponible a: <https://www.ethereum.org/> [Juny, 2019]

[8] The Hollywood Reporter: Entertainment Report

Disponible a :

[https://www.hollywoodreporter.com/sites/default/files/custom/Documents/PWC\\_chart\\_rev21012x5711px.pdf](https://www.hollywoodreporter.com/sites/default/files/custom/Documents/PWC_chart_rev21012x5711px.pdf) [Abril, 2019]

[9] Solidity - Solidity 0.5.3 documentation [online]

Disponible a: <https://solidity.readthedocs.io/en/v0.5.3/> [Febrer, 2019]

[10] SDLC Models Explained: Agile, Waterfall, V-Shaped, Iterative, Spiral [online]

Disponible a:

<https://medium.com/existek/sdlc-models-explained-agile-waterfall-v-shaped-iterative-spiral-e3f012f390c5> [Febrer, 2019]

[11] Jira | Software de seguimiento de proyectos e incidencias | Atlassian [online]

Disponible a: <https://es.atlassian.com/software/jira> [Febrer, 2019]

[12] Git [online]

Disponible a: <https://git-scm.com/> [Febrer, 2019]

[13] Quorum [online]

Disponible a: [www.goquorum.com](http://www.goquorum.com) [Juny, 2019]

[14] Apache Cordova [online]

Disponible a: <https://cordova.apache.org/> [Juny, 2019]

[15] MultiChain | Open source blockchain platform [online]

Disponible a: <https://www.multichain.com/> [Juny, 2019]

[16] Apache Solr - - Apache Lucene - The Apache Software Foundation! [online]

Disponible a: <http://lucene.apache.org/solr/> [Juny, 2019]

[17] ngX-Rocket [online]

Disponible a: <http://ngx-rocket.com> [Juny, 2019]

[18] Documentation | NestJS - A progressive Node.js web framework [online]

Disponible a: <https://docs.nestjs.com/> [Juny, 2019]

[19] The Best APIs are Built with Swagger Tools | Swagger [online]

Disponible a: <https://swagger.io/> [Juny, 2019]

[20] Sweet Tools for Smart Contracts | Truffle Suite [online]

Disponible a: <https://truffleframework.com> [Juny, 2019]

[21] json-path-value - npm [online]

Disponible a: <https://www.npmjs.com/package/json-path-value> [Juny, 2019]